



DEGREE PROJECT, IN SPEECH AND MUSIC COMMUNICATION FOR MASTER
DEGREE IN MEDIA TECHNOLOGY , SECOND LEVEL
STOCKHOLM, SWEDEN 2014

A Mobile Application for Improving Running Performance Using Interactive Sonification

JOEL FORSBERG

KTH ROYAL INSTITUTE OF TECHNOLOGY

COMPUTER SCIENCE AND COMMUNICATION (CSC)

A Mobile Application for Improving Running Performance Using Interactive Sonification

Abstract

Apps that assist long-distance runners have become popular, however most of them focus on results that come from calculations based on distance and time. To become a better runner, an improvement of both the body posture and running gait is required. Using sonic feedback to improve performance in different sports applications has become an established research area during the last two decades. Sonic feedback is particularly well suited for activities where the user has to maintain a visual focus on something, for example when running. The goal of this project was to implement a mobile application that addresses long-distance runners' body posture and running gait. By decreasing the energy demand for a specific velocity, the runner's performance can be improved. The application makes use of the sensors in a mobile phone to analyze the runner's vertical force, step frequency, velocity and body tilt, together with a sonification of those parameters in an interactive way by altering the music that the user is listening to. The implementation was made in the visual programming language Pure Data together with MobMuPlat, which enables the use of Pure Data in a mobile phone. Tests were carried out with runners of different levels of experience, the results showed that the runners could interact with the music for three of the four parameters but more training is required to be able to change the running gait in real-time.

En mobil applikation för att förbättra löpningsprestation genom interaktiv sonifiering

Sammanfattning

Det har blivit populärt med appar som riktar sig till långdistanslöpare, men de flesta av dessa fokuserar på resultat som kommer från uträkningar av distans och tid. För att bli en bättre löpare krävs att man förbättrar både sin kroppshållning och sin löpstil. Det har blivit ett etablerat forskningsämne under de senaste årtiondena att använda sig av ljudåterkoppling för att förbättra sin prestation inom olika sporter. Detta lämpar sig väl för aktiviteter där användaren behöver fokusera sin blick på något, till exempel under löpning. Målet med det här projektet var att implementera en mobil applikation som riktar sig till att förbättra långdistanslöparens kroppshållning och löpstil. Genom att minska på energin som krävs för att springa med en viss hastighet kan löparens prestationsförmåga öka. Applikationen använder sig av sensorerna i en mobiltelefon för att analysera användarens vertikala kraft, stegfrekvens, hastighet och kropps lutning genom att sonifiera dessa parametrar på ett interaktivt sätt där musiken som användaren lyssnar på ändras på olika sätt. Implementeringen gjordes i det visuella programmeringsspråket Pure Data tillsammans med MobMuPlat, som gör att implementeringen kan användas i en mobiltelefon. Tester genomfördes med löpare med olika grader av erfarenhet, resultaten visade att löparna kunde interagera med musiken för tre av de fyra parametrarna men mer övning krävs för att kunna förändra löpstilen i realtid.

Table of Contents

1	Introduction.....	1
1.1	Background.....	1
1.2	Problem.....	1
1.2.1	Goals.....	2
1.2.2	Limitations.....	2
1.3	Thesis Contents.....	2
2	Theory	3
2.1	The Physics of Running.....	3
2.1.1	Running Gait Cycle.....	3
2.1.2	Biomechanics and Running Technique.....	3
2.1.3	Running Economy.....	4
2.2	Sensors in Mobile Devices.....	5
2.3	Sonification.....	6
2.3.1	The Concept of Sonification.....	6
2.3.2	Sonification Methods.....	7
2.3.3	Information Through Sound.....	7
3	Related Work	9
4	Method	11
4.1	Pure Data and MobMuPlat.....	11
4.2	User Interface.....	13
4.3	From Sensor Data to Running Gait Analysis.....	14
4.3.1	Overview.....	14
4.3.2	Introduction to the Sensor Data.....	14
4.3.3	Vertical Force.....	17
4.3.4	Step Frequency.....	20
4.3.5	Velocity.....	21
4.3.6	Body Tilt.....	22
4.4	Sonifying the Body Movements.....	23
4.4.1	The Music Player.....	23
4.4.2	Mappings.....	23
4.4.3	Time Stretching and Pitch Shifting.....	26
4.4.4	Filters.....	27
4.4.5	Auditory Icons.....	27
4.5	Sound Examples.....	28

5	Results	29
5.1	Validation of the Step Detection.....	29
5.2	Experiments	29
5.2.1	Vertical Force	30
5.2.2	Step Frequency	32
5.2.3	Velocity	33
5.2.4	Body Tilt.....	35
5.2.5	Perception of Combined Parameters	37
6	Discussion.....	38
7	Conclusions and Future Work.....	40
7.1	Conclusions.....	40
7.2	Future Work.....	40
	Bibliography	42

1 Introduction

This chapter presents a background for the topic of sonic feedback for running and its possibilities. The project's goals, delimitations and the thesis contents are outlined.

1.1 Background

There are hundreds of applications that give feedback to long-distance runners. Most existing applications focus on results that are derived from distance and time (Edelstam & Eelde 2013). The user's running gait, which is a key for improving the running performance, is not addressed in most commercial applications. Along with the development of applications for running, an increasing amount of sensors that can be connected to mobile phones via Bluetooth or Wi-Fi have emerged. Most modern mobile phones come also with internal sensors, such as accelerometers, gyroscopes and magnetometers. When using sensor data from for example an accelerometer and a gyroscope that are placed on a strategic part of the runner's body, real-time information about the running gait can be extracted and be used to give feedback so that the body movement and running technique of the user can be adjusted accordingly.

Sonification, or sonic feedback, suits well for purposes like these because the visual focus can be maintained on the surrounding environment or the track while taking part of the auditory information (Kramer et al. 1999). Information about the runner's vertical displacement, step frequency and body tilt are examples of what the runner can take part of in order to adjust them for a more efficient running technique.

Pure Data (Pd) (IEM 2014) is a visual programming language that is frequently used in sonification research, and is well suited for implementations like these as it can be used on a mobile device together with e.g. Mobile Music Platform (MobMuPlat) (Iglesia n.d.), PdDroidParty (McCormick et al. 2013) or previously RjDj (Reality Jockey Ltd. 2013). The sensor data can be used in the Pd implementation for changing and adding sounds in the music to which the runner is listening. As it is common for runners to listen to music, sonification of this kind can easily be made a part of their exercise.

1.2 Problem

The main problem addressed in this project was how running technique might be improved in real-time when using sonification in an interactive way with the user. The task was to implement a program that can be used to sonify data from the onboard sensors of a mobile phone that is placed on a runner's body. Sensor data from a smartphone was used to retrieve information about the runner's body posture and running gait in order to sonify it to give feedback to the runner so that the running technique can be adjusted accordingly. The sonification should let the user know if the running gait has been corrected solely by listening to it. Hence, a functional mapping between sensor data and sound is critical for the application to work in practice.

1.2.1 Goals

The goals of the project were to:

- Implement a mobile application with interactive sonification for running using data from the internal sensors of a mobile phone placed on the user's body to sonify the user's movement
- Test the application on runners with different levels of experience to see if running performance can improve, as a consequence of a better running technique that leads to less energy consumption, through interactive sonification

1.2.2 Limitations

Since a good running technique has many different parameters that need to be monitored, sensor data from more than one part of the runner's body would be needed to get a complete picture of the running gait. The design decision of using a mobile phone and its onboard sensors alone, placed on the lower back of the user, excluded some aspects of the running technique. More sensors, e.g. placed on the runner's feet, could be added to get more information about the running technique.

1.3 Thesis Contents

The structure of the thesis is as follows. This introductory chapter gave a background for the thesis and presented the problem together with goals and limitations. Chapter 2 provides the science behind a good running technique and introduces sensors that can be used to analyze running gait and how sonification should be done in the most natural way for users to understand. In chapter 3, related research is presented to show what has previously been done in the area of sonification in relation to sports. Chapter 4 goes through the method of the project with an explanation of the implementation and the chosen sonification method. The test results are presented in chapter 5. Then follows a discussion of the project in chapter 6. Conclusions are drawn in chapter 7 together with suggestions for future work.

2 Theory

A walkthrough of the physics and biomechanics of running; sensors that can be used for extracting data about the runner with mobile phones are introduced; the concept of sonification and how sounds can be used for conveying information are presented.

2.1 The Physics of Running

2.1.1 Running Gait Cycle

The repetitive nature of running is an important part of analyzing the running technique. Novacheck (1998) wrote about the running gait cycle, which starts when one foot gets in contact with the ground and ends when the same foot is back on the ground again. The steps of the cycle are listed below:

- *Initial contact (IC)*
- *Stance phase reversal (StR)*
- *Toe off (TO)*
- *Swing phase reversal (SwR)*
- *Initial contact, repeat*

During the steps listed above there are two phases, which are presented by Novacheck as *absorption* and *generation*. The absorption phase occurs from SwR, through IC, to StR and the generation phase occurs from StR, through TO, to SwR. A concluding figure of all the phases can be seen in Figure 1. Novacheck wrote that there are no periods when both feet are in contact with the ground during running. When the runner is moving faster, less time is spent in stance.

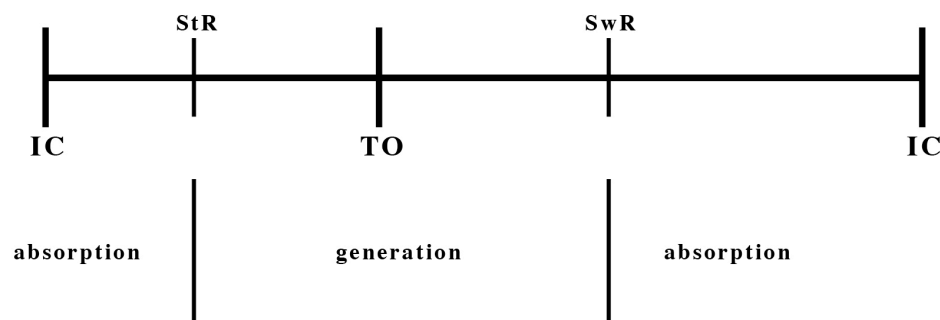


Figure 1. The running gait cycle, which starts when one foot gets in contact with the ground (IC) and ends when the same foot is back on the ground again. A phase called absorption phase occurs between swing phase reversal (SwR) and stance phase reversal (StR) and the generation phase occurs between StR and SwR.

2.1.2 Biomechanics and Running Technique

A good running technique is learnt in three stages, according to Tucker (2007). He claimed that the first learning phase comes naturally, then the running is refined through practice. Through instruction, subtle changes in the technique can be taught. Tucker wrote

about the two major running techniques that have emerged, the Pose and Chi running methods, claiming that the two are basically the same idea packaged with different names.

Gonowon (2007) presented which external forces a runner is exposed to when running and what the effects of those are, which are listed below:

- *Drag force* works against the forward motion of the runner
- *Gravity* pulls the runner towards Earth
- *Normal force* prevents the runner from falling through the ground
- *Friction* allows the foot to grip the ground for balance

For the human body, the center of gravity is equal to the center of mass, which is in front of the spine below the navel (Gonowon 2007). Tucker (2007) referred to the hips as the center of mass for the runner. According to the Pose running method and basic physics, the runner should lean forward as a straight line from the ankles and upward, then the gravity force pulls the runner forward (Tucker 2007; Gonowon 2007). Tucker claimed that the hips of the runner should be as far forward as possible. Novacheck (1998) wrote that the center of mass is lowered when going from walking to running, as a consequence of the forward tilting. When runners are leaning forward more they can run faster because the gravity force pull down the runner in a horizontal forward direction, instead of downward (Gonowon 2007). However, if only the runner's upper body is bent forward, the center of mass is moved and the running becomes less efficient.

The landing point of the foot should be as close as possible to the ground projection of the center of mass (Tucker 2007; Gonowon 2007), Gonowon wrote that if the landing of the foot is in front of that point, the force is working opposed to the forward direction which causes the runner to slow down. She wrote that during running the arms should be relaxed and swung naturally. She also claimed that by shortening the arms, e.g. by having the arm bent 90 degrees or less, it is possible for the runner to run faster.

As to whether the runner should land on the forefoot or the rearfoot, there are mixed opinions. Novacheck (1998) wrote that about 80 % of long-distance runners land with their rearfoot and claimed that generally as running speed increases, the initial contact between the foot and the ground goes from the rearfoot to the forefoot, this can be seen as the distinction between running and sprinting. Tucker (2007) wrote that 75 % of elite runners land on their heels, but according to the Pose method runners should land on their forefoot. Tucker concludes that where on the foot the landing takes place is not very important.

2.1.3 Running Economy

The running performance depends on the runner's energy consumption. Conley & Krahenbuhl (1980) wrote that "running economy is considered to be the steady-state oxygen consumption for a standardized running speed". According to Saunders et al. (2004) running economy (RE) is the energy demand for a given velocity when running. They claimed that runners with good RE use less energy than runners with bad RE, which leads to a better performance for the ones with good RE. For highly trained runners, Conley & Krahenbuhl wrote that variations in performance to a high degree depends on RE. Prampero et al. (1993) presented that the maximal metabolic power a runner can maintain in running is a function of oxygen consumption, and Williams & Cavanagh (1983) wrote that the metabolic cost is related to the mechanical cost of the runner in order to move forward. The runners' step frequency is regarded as a factor for affecting the mechanical cost, since the gravity force needs to be worked against a larger number of

times when having a high step frequency (Eriksson & Bresin 2010), although most elite runner's have a higher step frequency than the average recreational runner (Phillips 2013). By having a consistent and correct step frequency, an optimized RE is easier to get (Bood et al. 2013).

RE is influenced by both biomechanical and physiological factors such as core temperature, heart rate and ventilation (Saunders et al. 2004). Saunders et al. showed that long-distance runners have better RE than middle-distance runners because of the smaller vertical displacement of the runners' center of mass, they also stated that lowering the vertical oscillation of the body's center of mass is a key factor for improving ones RE.

2.2 Sensors in Mobile Devices

To measure the running gait of a runner, sensors can be put on the runner's body to get data about e.g. body angles and accelerations. In this section, both onboard sensors of mobile phones and three examples of external sensors that can be connected to mobile phones are introduced.

Most modern mobile phones feature an accelerometer, gyroscope and a location tracking system through GPS. Apple's iPhone has from the fourth generation and later included a 3-axis accelerometer, which measures acceleration in the X , Y and Z direction. There is also a gyroscopic sensor in the iPhone, which measures the rate of rotation in three dimensions called *roll*, *pitch* and *yaw*. The six dimensions of the accelerometer and gyroscope are shown in Figure 2.

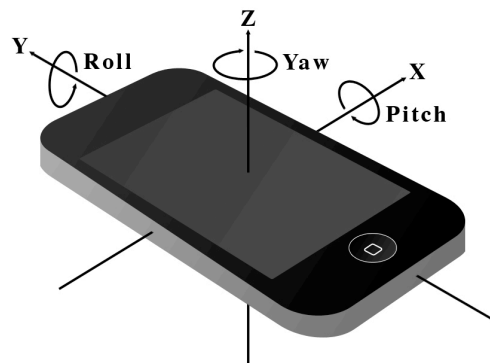


Figure 2. Dimensions from accelerometer and gyroscope measurements. Accelerometer dimensions are X , Y and Z and gyroscope dimensions are pitch, roll and yaw.

It is also possible to connect external sensors to mobile phones. There is e.g. the CC2541 SensorTag, which is the name of a development kit from Texas Instruments (2013). It makes use of Bluetooth low energy to send data from six sensors, listed below:

- IR temperature sensor
- Humidity sensor
- Pressure sensor
- Accelerometer
- Gyroscope
- Magnetometer

From x-io Technologies (2013) comes x-OSC, which sends data over Wi-Fi using the Open Sound Control (OSC) format. OSC is a protocol for communication that is

optimized for modern network technology, it was originally developed at UC Berkeley Center for New Music and Audio Technology (CNMAT 2011). Three sensors have been put on the x-OSC board, listed below:

- Accelerometer
- Gyroscope
- Magnetometer

Another alternative comes from Notch, which makes it possible to connect up to ten sensors placed on the body that also can give haptic feedback through their vibration motor (Notch Interfaces 2014). The sensors send data with Bluetooth low energy to a smart device and allow for a skeletal tracking of the user's body.

2.3 Sonification

2.3.1 The Concept of Sonification

The history of sonification started before there was a word for it. A well-known example of using sound to convey information is also one of the most successful ones, i.e. the Geiger counter from the early 1900s (Kramer et al. 1999). The Geiger counter is used to measure radiation and makes the sound of clicks with a frequency that is proportional to the radioactivity (Worrall 2009). Today, sonification is an established research area and the following definition seems to have been agreed upon within the research community (Dubus & Bresin 2013):

“Sonification is defined as the use of nonspeech audio to convey information. More specifically, sonification is the transformation of data relations into perceived relations in an acoustic signal for the purposes of facilitating communication or interpretation.” (Kramer et al. 1999)

Although this definition is commonly used, there are discussions on where to draw the line for sonification, according to Dubus & Bresin (2013). Worrall (2009) questioned why speech audio has to be excluded from the definition and presumed it was to discriminate between sonification methods and e.g. text-to-speech software and other speech-related areas.

Hermann & Hunt (2005) wrote that sonification is the use of sound to present information to users so that they can get a deeper understanding of data or a process by listening. Since sonification can be used with any kind of data, interactively or not, it has big possibilities (Dubus & Bresin 2013). It is particularly useful when the user is not facing in a specific direction, unlike e.g. visual display, according to Kramer et al. (1999), who also wrote that the auditory perception of human beings is sensitive to the temporal characteristics of sound. Hence, sonification suits well for time-related tasks, in particular for the perception of the user's body motion (Dubus & Bresin 2013). Schaffert et al. (2009) agreed upon that, claiming that movement and sound are naturally bound together and due to the high temporal resolution of human hearing, it is possible to hear specific information about timing and movement in order to synchronize with the sound. Hunt & Hermann (2004) claimed that interaction is important in sonification since it is the way nature works, the world reacts when the human acts. Later, they wrote that interactive sonification can be described as information through sound about data in order for the users to refine their activity (Hermann & Hunt 2005).

2.3.2 Sonification Methods

Sonification can be done in a number of different ways. Worrall (2009) divided different sonification methods into three types of data representations, listed below:

- *Discrete data representations*
- *Continuous data representations*
- *Interactive data representations*

A *discrete data representation* is when every data point is sonified with an individual auditory event, which works in a symbolic manner (Worrall 2009). *Auditory icons* is an ecological approach to exemplify this, Dubus & Bresin (2013) described the technique as making use of metaphorical perceptual relationships from environmental sounds. A well-known example of an auditory icon is the sound of paper that is being crumpled and thrown to symbolize the deletion of a file in the computer (Susini et al. 2012). Dubus & Bresin described a similar technique without the metaphorical aspects in the sound, which is to use *earcons* with synthetic sounds, but then the meanings have to be learned by the user beforehand.

Continuous data representations are used for exploring data in order for the user to learn more about the system that produced it (Worrall 2009). *Audification* is an example of this type of representation, which is a direct playback of the data as sound waves, as described by Dubus & Bresin (2013). They also proposed that *parameter mapping sonification* can be used continuously, where the data is mapped to various auditory dimensions.

Interactive data representations can be e.g. *model-based sonification*. This technique is described as a virtual sounding object that is created when data are added to it and the sound triggers in the interaction between the user and the system (Dubus & Bresin 2013).

2.3.3 Information Through Sound

Sound is used in a wide variety of products. An important aspect is the relationship between the sound and the information to be conveyed to the user (Susini et al. 2012). According to Susini et al., mapping sound with information can have a symbolic, iconic or causal meaning, where the last two rely on knowledge the user already have whilst a symbolic sound has to be explained beforehand. Suied et al. (2010) investigated auditory warnings to alert users of potential danger or information arrival and how users perceived the warnings. They wrote that the perceived urgency of an auditory warning depends on the pitch, i.e. the higher the pitch, the higher the perceived urgency. According to Suied et al., the user's response time becomes shorter when using everyday sounds instead of artificial auditory warnings because the user understands them more quickly. Suied et al. also showed that white noise that was modulated with the temporal envelope of animal sounds resulted in a similar response time as for the animal sounds themselves, proposing that the acoustic difference is more important than the semantic or cognitive difference.

The naturalness of the sonic feedback is important to natural interaction (Rocchesso et al. 2009), because the user is accustomed to how sound behaves in the physical world (Hermann & Hunt 2005). An acoustic event can have several different attributes at once, e.g. pitch, modulations, amplitude envelope over time, spatial location, timbre and brightness (Hermann & Hunt 2005). Dubus & Bresin (2013) investigated how physical quantities have been mapped to auditory dimensions in past studies. The investigated physical quantities were kinematics, kinetics, matter, time and dimensions (i.e. geometry of objects and spaces), and the investigated auditory dimensions were pitch-related, timbral, loudness-related, spatial and temporal. Listed below are the results of the study,

showing which auditory dimensions that have been most or least used for each physical quantity:

- *Kinetics*: Spatial auditory dimensions were less used than other dimensions.
- *Kinematics*: Pitch-related and temporal auditory dimensions were used more than loudness-related dimensions.
- *Matter*: Spatial auditory dimensions were not used at all.
- *Time*: Timbral auditory dimensions were used more than loudness-related dimensions. Spatial auditory dimensions were used less than pitch-related, temporal and timbral dimensions.
- *Dimensions*: Loudness-related auditory dimensions were used less than pitch-related and timbral dimensions. Spatial auditory dimensions were used less than pitch-related, temporal and timbral dimensions.

Dubus & Bresin (2013) also looked at the horizontal and vertical physical dimensions, and found that spatialization has been most used for the horizontal dimension and pitch has been most used for the vertical dimension.

3 Related Work

This chapter gives a review of previous work in the area of interactive sonification and real-time feedback for various user tasks. There is also an introduction to commercial mobile applications that give feedback to long-distance runners.

The research area of sonic feedback for sports applications is well established, and a number of prototypes and applications that can be related to this project have been developed. In the context of outdoor activities, Barrass et al. (2010) investigated six interactive sonifications of accelerometer data. They used an iPod Touch from Apple for synthesized sonification in real-time of the onboard accelerometer. The participants did not perform a specific task and the goal was to investigate how sonification could be used in different situations, such as walking, jogging, martial arts, yoga and dancing. Barrass et al. implemented the sonifications using Pd together with the RjDj software. Of the six sonification methods used, the most preferred by the participants was called algorithmic music, where the X, Y and Z data from the accelerometer controlled three instruments based on FM synthesis. The authors described the audio output to sound like “esoteric, generative or improvisational ambient electronic music”. In another study, Varni et al. (2012) investigated the possibility for users to synchronize their gestures using sonification from acceleration data measured by a mobile phone. It was tested with three different sonification methods that depended on the movement. The first method applied a non-linear filter to the music, the second added and removed instruments from a multi-track musical piece and the last one changed the performance parameters of the music. The authors suggested that the sonification helped for longer synchronization times but the participants perceived the situation without sonification as an easier task.

Hummel et al. (2010) conducted a study to show that a performer who is making acrobatic movements on a German wheel can improve the performance when using sonification. They used sensor data from a magnetometer and proposed four methods for sonification, eventually testing two of them on a performer. The best method was concluded to be an event-based one, which generated a sound each time certain conditions were fulfilled, i.e. the circle was divided into a number of steps and the sound was generated step-wise as the wheel was rolling. Furthermore, an implementation for a real-time sonification system that aimed to improve the exercise of doing biceps curls was done by Yang & Hunt (2013). The results indicated improvement of the movement quality but no clear improvement in the physical range of movement was seen.

Dubus (2012) presented four sonification methods in a mobile application for elite rowers. According to the rowers, the most pleasant method of the four was a Pd implemented patch with synthesized wind sound corresponding to the velocity of the boat. The idea behind it was an ecological approach, i.e. the wind is perceived as louder when moving faster, thus giving a natural feedback to the rowers. In the same context, Schaffert et al. (2009) explored different sonification methods for rowing movement. Several questions were raised regarding functionality and aesthetics of sonification and the relation between them. Later, Schaffert et al. (2010) proposed a potential version of sonic feedback for rowers and tested it on elite junior athletes. The sonification focused on the acceleration data, which were mapped to tones on the musical tone scale. This means that when the acceleration of the boat increased, the pitch of the tone was higher and vice versa. According to the athletes, this mapping was intuitive and it resulted in an increase of the boat velocity.

A study by Eriksson & Bresin (2010) investigated the possibility of using an external sensor for giving auditory feedback to a runner in real-time. They implemented a system using a sensor together with a mobile phone using Java ME. After computing the vertical displacement and step frequency of the runner, sonic feedback was given so the runner could adjust accordingly. In his master's thesis, Bolibar (2012) wrote about sonic feedback for long-distance runners using a *Kinect* to analyze the running technique. Data from the *Kinect* was sent to a computer where a Pd implemented program was used for sonification of the runner's vertical displacement, tilt and step distance. The running was done on a treadmill since the relation between the runner and the *Kinect* had to be constant. Bood et al. (2013) conducted an experiment about coupling running cadence to music to examine the physical and perceived effects of exertion. They showed that the cadence was more consistent when running with a metronome than with motivational music and that the time to exhaustion was longer when the runner listened to music (or a metronome) than when having no acoustic stimulus. Bood et al. also wrote that when listening to music while running, the perceived exertion can be reduced by up to 10 % during a low-to-moderate level of physical exercise. Crowell et al. (2010) showed that runners could use real-time visual feedback from an accelerometer to reduce tibial acceleration and vertical force loading rates. They claimed that by doing this, the risk for stress fractures would be reduced.

In their Bachelor thesis, Edelstam & Elde (2013) wrote about the mapping between input and output in applications that give sonic feedback for runners. They proposed to use the music that the runner is listening to as a starting point, and then change or add other sounds to the music. They also wrote about how twelve existing mobile applications for running feedback were designed. The applications that they analyzed were Endomondo, Runkeeper, Garmin Fit, Nike+ Running, SportyPal, Runtastic, Cardiotrainer, Runmeter, iRunner, MapMyRun, STT Sports Tracker and Google My Tracks. All applications were focused around tempo, time and other geographical data; one or more also included pulse, hydration and oxygen consumption. Voice feedback regarding instantaneous speed and splits (parts of a distance) was possible in all except one of the applications. Other commercial mobile applications for running feedback that can be mentioned are e.g. PUMATRAC, Strava Run and Adidas MiCoach.

4 Method

This chapter describes how the application was implemented and why the particular design choices were made. An introduction to the platform that was used to implement the application is also provided.

In order to improve the running performance through a better RE, four properties of the running technique were selected based on the theory presented in chapter 2:

- *Vertical displacement*, in the application referred to as *vertical force*, by lowering the vertical force and thereby the vertical displacement the RE can be improved
- *Step frequency*, e.g. by lowering or having a consistent step frequency the RE can be optimized, while an increment may be desired for casual runners
- *Velocity*, a central aspect in running for both casual and elite long-distance runners
- *Body tilt*, by tilting the body forward more an increased velocity is achieved

The above-mentioned properties will be more thoroughly explained in the coming sections together with the mappings to sound.

4.1 Pure Data and MobMuPlat

The application was implemented using Pd together with MobMuPlat. Pd is an open source visual programming language that is used for generating and processing audio, video, 2D/3D graphics, MIDI and more (IEM 2014). It consists of two versions, the original Pd-vanilla written by Miller Puckette and Pd-extended which comes with extra libraries written by the Pd community. A program, also known as a patch, in Pd is implemented using “cables” to connect *objects* and *messages* instead of writing lines of codes. It is reminiscent of how audio can be connected through cables in a real world situation and is therefore an intuitive way to work with audio. In Figure 3, a simple “hello world” example implemented in Pd-extended is shown, using only Pd-vanilla objects, where the message “hello world” is sent to the *print* object.



Figure 3. "Hello world" in Pd-extended.

Together with the MobMuPlat application (Iglesia n.d.), it is easy to use a Pd patch in a mobile iOS device with a friendlier graphical user interface (GUI). MobMuPlat was implemented using libpd, a tool which makes it possible to embed Pd patches in other programming languages (Kirn 2012). When implementing the application with MobMuPlat, MobMuPlatEditor is used to make the GUI, which sends messages to the Pd

patch. To demonstrate the workflow of implementing a mobile application with MobMuPlat and Pd, a “hello world” example is shown in Figure 4 and Figure 5.

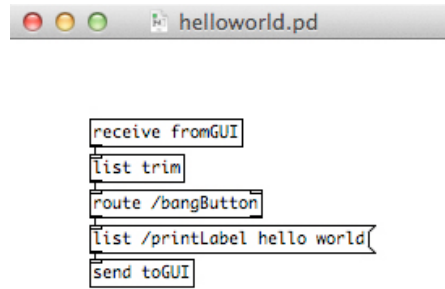


Figure 4. "Hello world" in Pd with MobMuPlat from the Pd point of view.

Figure 4 shows how the Pd patch receives a “bang” from the button when it is pressed and released in the GUI, which has been given the address `/bangButton` in MobMuPlatEditor in Figure 5. The “bang” pushes the message “hello world” to the GUI, where the text field that is showing the text has been given the address `/printLabel`.

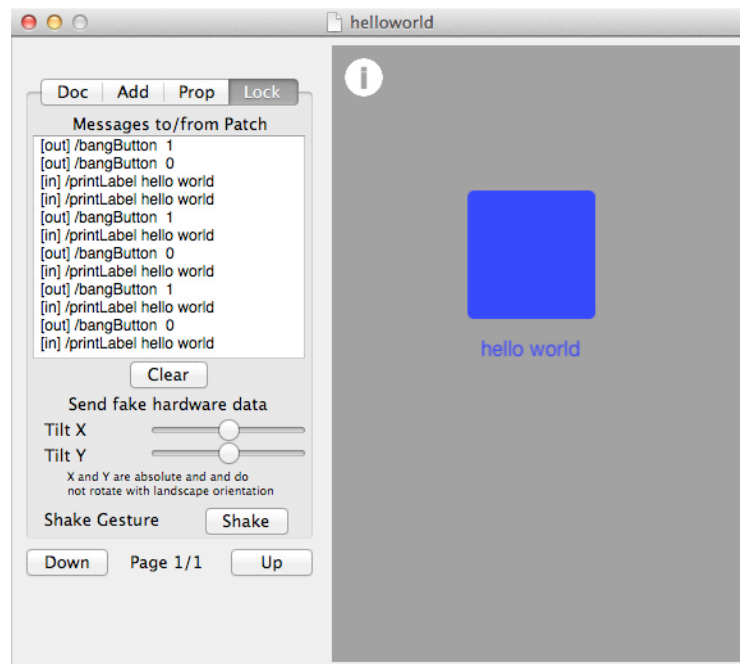


Figure 5. "Hello world" in Pd with MobMuPlat from the MobMuPlatEditor point of view.

When running this on an OS X computer, a patch called *PdWrapper.pd* has to be running in the background. This patch uses OSC to simulate the communication between the GUI and the Pd patch that happens internally in the iOS application when using it on a mobile device. A special distribution of Pd-vanilla has been provided by Iglesia Intermedia and can be downloaded from the MobMuPlat website, it includes some extra objects to make the OS X simulation possible (Iglesia n.d.).

To run the program on an iOS device, MobMuPlat has to be downloaded from App Store to the device. Then the file containing the GUI from MobMuPlatEditor together with the Pd patch and any additional files have to be copied to the MobMuPlat application’s documents in iTunes.

4.2 User Interface

The aim of the application was to make an as user-friendly experience as possible by letting the sonic interaction play the biggest role. Consequently, the goal was to decrease the importance of a GUI by having the user interact as little as possible. The only tangible contact the user should have with the device and the GUI is when setting the desired improvement of each parameter before the running exercise. This is designed to be as easy as possible with toggles for choosing which parameters to activate and sliders for adjusting them. An on/off button has to be pushed before putting the device on the body and starting to run. During the first 30 seconds of running, average values of all the parameters are computed. After those 30 seconds the sonification begins and the user should immediately know if the running is “correct”. A statistics page was created to show the results of the training. The page shows the average values from the last exercise, together with its time and the number of steps. All the pages of the application are shown in Figure 6.

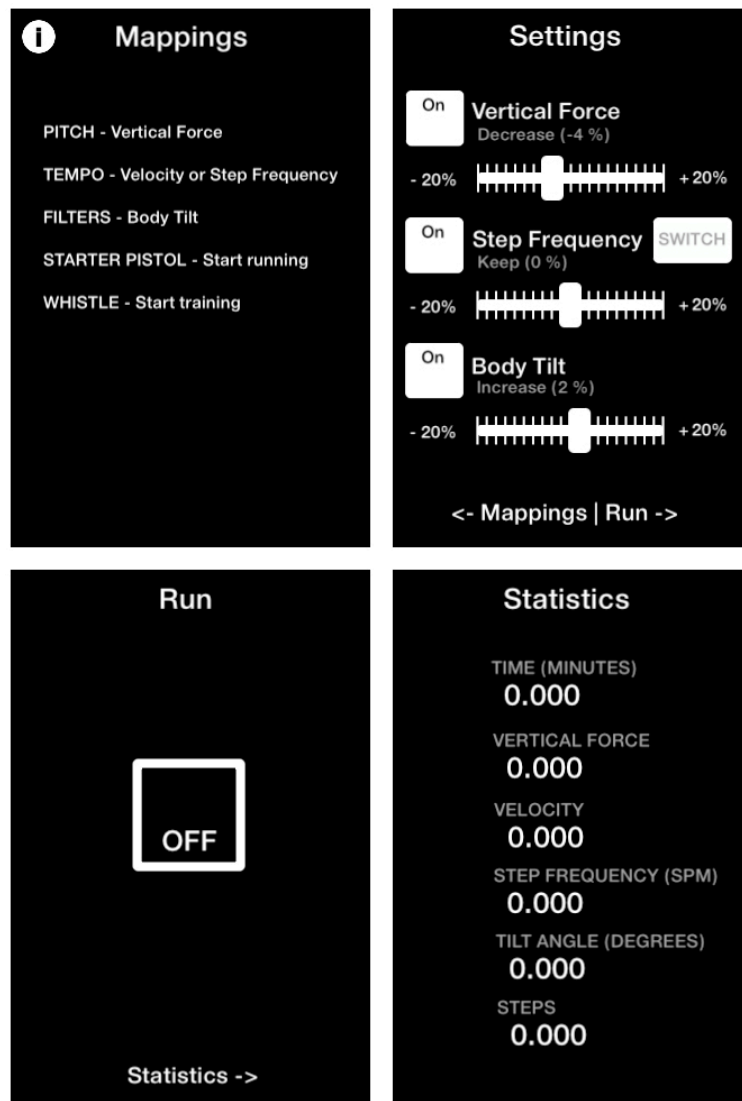


Figure 6. The four pages of the GUI. The “Settings” page is the initial page and lets the user set the desired improvements for the chosen parameters before sliding to the “Run” page and starting the application. The different mappings are presented on the “Mappings” page. After the running, statistics from the exercise are shown on the “Statistics” page.

The application was implemented on an iPhone 4S that should be placed on the user's lower back, close to the center of mass, in a belt from Adidas called *Adidas Media Belt X14341*.

4.3 From Sensor Data to Running Gait Analysis

4.3.1 Overview

The input values from the GUI are received through the *receive fromGUI* object. Each slider and toggle in the GUI has been given an address, which is used to route the messages to different outlets with the *route* object. The data from the iPhone is received from the *receive fromSystem* object and sent to subpatches to compute the different parameters that control the sonification. This will be described more thoroughly in the following sections. The main Pd patch is shown in Figure 7.

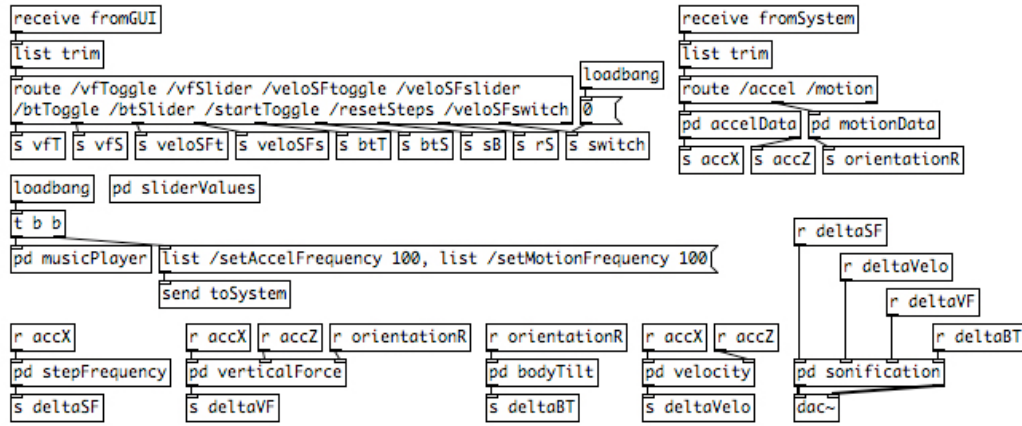


Figure 7. An overview of the main Pd patch, which shows how the data are received from the sensors and the GUI before being sent to the different subpatches for calculating the parameters that are used for the sonification.

4.3.2 Introduction to the Sensor Data

The Pd patch receives the sensor data from the object *receive fromSystem* and routes the acceleration data, */accel*, and the device motion data, */motion*, to different outlets as shown in Figure 8. The left outlet outputs the three axes (X, Y and Z) from the accelerometer and the right outlet outputs roll, pitch and yaw from the pre-processed device motion data, which originates from the accelerometer, gyroscope and magnetometer.

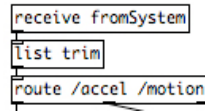


Figure 8. Routing of the received acceleration and device motion sensor data.

The unit of the accelerometer is g, i.e. 1 g is approximately equal to 9.81 m/s^2 . If the phone is held still, the gravitational force will make the accelerometer output a value ranging from negative one to one depending on the phone's orientation. The output of the device motion data is an angle between negative π and π , based on the phone's current orientation in space. The update frequency for both the acceleration and device motion

data was set to 100 Hz through the message that is shown in Figure 9, which is sent every time the patch is opened.

```
list /setAccelFrequency 100, list /setMotionFrequency 100[
send toSystem
```

Figure 9. Setting the update frequency of acceleration and device motion data.

The patch uses the Z-axis and X-axis of the accelerometer because of how the phone is placed in the belt. The phone's orientation can be seen as horizontal when placed in the belt, i.e. the X-axis is perpendicular to the surface of the Earth. A subpatch was made to unpack the accelerometer data and send it through if the user has turned on the application, which is shown in Figure 10. The *r sB* object receives a one or a zero depending on the on/off button in the GUI.

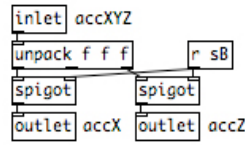


Figure 10. Unpacking the accelerometer data.

An example of the raw sensor data that the Pd patch receives during running is shown in Figure 11. The acceleration goes up and down during the running gait cycle and due to the gravitational force, the oscillation is not centered around zero.

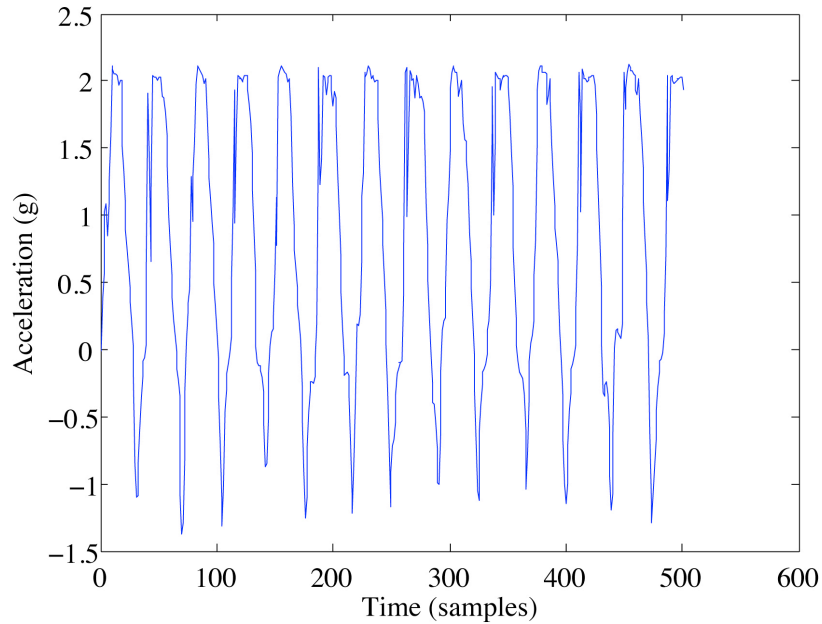


Figure 11. Example of the acceleration data from the X-axis during running.

The Z-axis from the acceleration data is shown in Figure 12. The oscillating movement can still be seen in the Z-axis acceleration but it is not as distinguished as in the up-down acceleration from the X-axis.

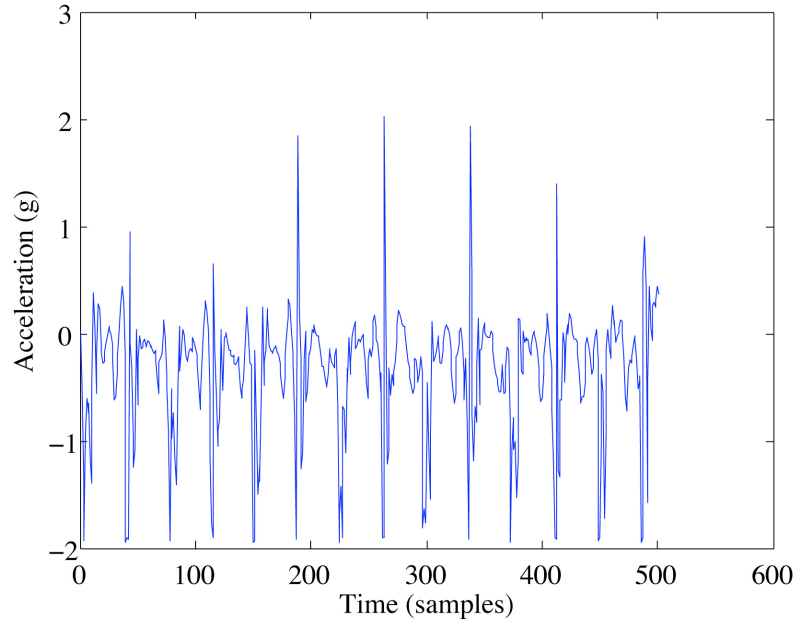


Figure 12. Example of the acceleration data from the Z-axis during running.

From the device motion data, the roll angle is used. A subpatch was made to unpack the incoming device motion data and convert it. Instead of ranging from negative π to π , it is converted so that it ranges from zero, when the X-axis is parallel to the surface of the Earth, to one, when the X-axis is perpendicular to the surface of the Earth. This conversion was calculated with the formula in equation 1:

$$\Phi = 1 - \frac{2 \left| \phi - \frac{\pi}{2} \right|}{\pi} \quad (1)$$

where ϕ is the angle received from the sensor data and Φ is the converted value between one and zero. The subpatch that unpacks and converts the device motion data is shown in Figure 13.

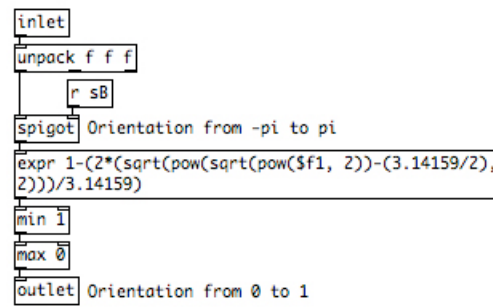


Figure 13. Unpacking and converting the device motion data to a value between zero and one depending on the phone's orientation in space.

An example of the converted value is shown in Figure 14. When attaching the phone in a position as described before, the roll angle can be regarded as the tilt of the body, where a value of one means that the user runs with the body perpendicular to the surface of the Earth and a value of zero means that user runs with the body parallel to the surface of the Earth.

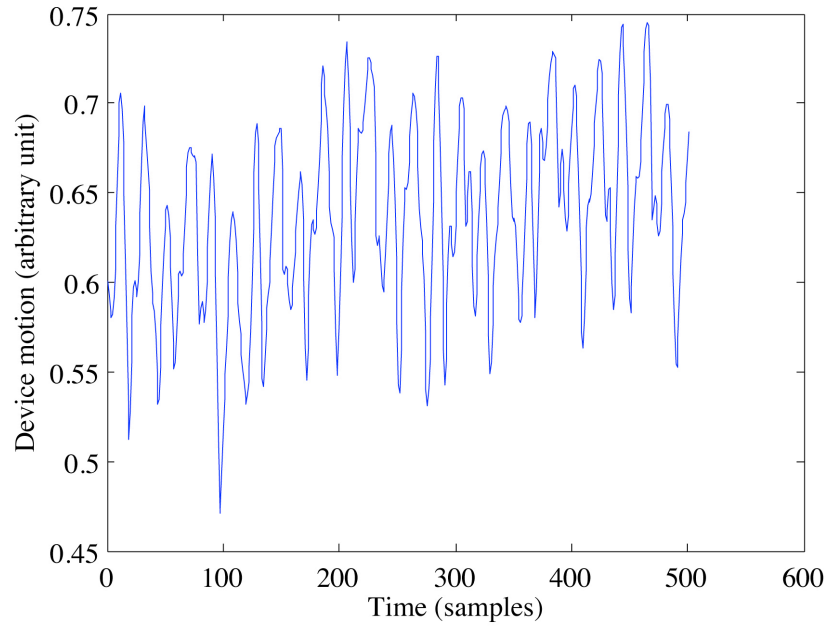


Figure 14. Example of the device motion data from the roll angle after conversion during running.

4.3.3 Vertical Force

The data from the accelerometer was used as a measurement of the force, and an indication of the vertical displacement. A definite value of the force was not interesting in this application since the measured acceleration only is compared to earlier values of the acceleration, as will be shown later. Together with the fact that acceleration has a direct proportionality with force according to Newton's second law of motion in equation 2, using the acceleration to get a hint about the force was regarded as enough:

$$F = ma \quad (2)$$

From here on when vertical force is mentioned, it really is vertical acceleration. The decision of naming it vertical force instead of vertical acceleration was made because the concept of vertical force is more established in the field of sports research. It was also thought to be easier to understand from a user perspective.

The vertical force is calculated as the acceleration perpendicular to the surface of the Earth. The acceleration from the X- and Z-axis is used together with the roll angle from the device motion data and sent to the subpatch *pd verticalForce*, which is shown in Figure 15.

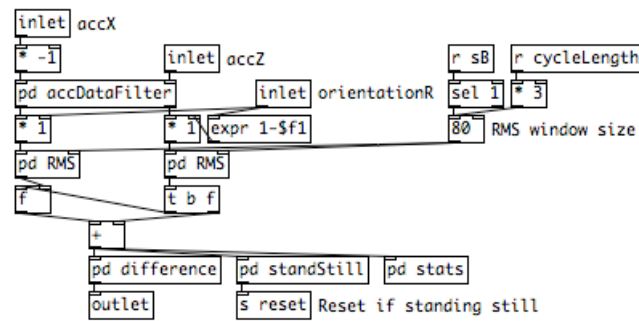


Figure 15. Calculating the vertical force with the X- and Z-axis from the accelerometer together with the roll angle from the device motion data.

In the *pd accDataFilter* subpatch, there is a low-pass filter and a high-pass filter. The low-pass filter is there to reduce noise from the incoming data stream and is shown in equation 3:

$$y_i = \alpha x_i + (1 - \alpha)y_{i-1} \quad (3)$$

where x is the input data, y is the filtered data and α is the filtering factor, set to 0.2. The high-pass filter is used to remove the constant acceleration from the gravitational force, which is shown in equation 4:

$$y_i = x_i - [\alpha x_i + (1 - \alpha)y_{i-1}] \quad (4)$$

where α is set to 0.5. The X-axis values are then multiplied with the roll angle from the orientation Φ . The Z-axis values are multiplied with $1 - \Phi$, where Φ is the roll angle to get the vertical force independent of the phone's orientation. Since the vertical acceleration is oscillating up and down, with negative force in the absorption phase of the running gait cycle and positive force in the generation phase, the root mean square (RMS) is used to measure the force of the oscillation during a decided interval n , as shown in equation 5:

$$x_{RMS} = \sqrt{\frac{\sum_{i=1}^n x_i^2}{n}} \quad (5)$$

where x is the data, both the X-axis and the data from the Z-axis is calculated as seen previously in Figure 15, where the RMS value is calculated in the *pd RMS* subpatch. The interval n is referred to as the RMS window size and is initially set to 80 samples but is then changed with values from the *pd stepFrequency* subpatch to have the window size equal to the time of six steps. When the window size is e.g. 80 samples, it means that a new value is output every 800 ms, since the update frequency is set to 100 Hz.

After the summation of the X- and Z-axis calculations, each calculated vertical force value is compared to the average value from the first 30 seconds of running after an eight second delay in the subpatch *pd difference* as shown in Figure 16. The delay is there so the user can start to run before the actual average calculation starts.

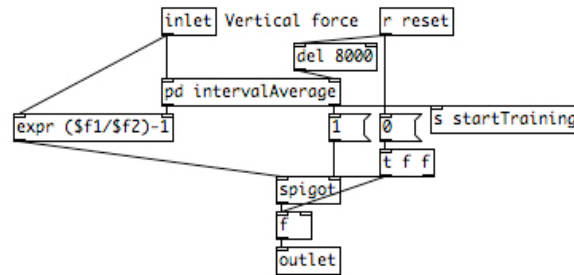


Figure 16. Comparing vertical force with the average value from the first 30 seconds of running.

As previously described, the user sets the desired improvement of each parameter in percentage before the running. This is taken into account when calculating the average value, e.g. if the user wants to reduce the vertical force with 4 %, the average value is multiplied with 0.96. In the subpatch *pd intervalAverage*, the desired vertical force is calculated during the first 30 seconds of running as shown in Figure 17.

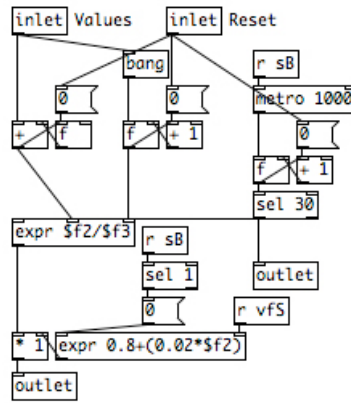


Figure 17. Calculating the average value from the first 30 seconds of running.

The *r vfS* object receives a value from the GUI slider, where the users choose how they want to change the vertical force. The *verticalForce* patch is also responsible for checking if the user stands still or not. This is done inside the subpatch *pd standStill* shown in Figure 18, which sends a bang with the variable name *reset* if the vertical force is less than 0.07 for eleven seconds. It also sends a one or a zero to the *pd stepFrequency* subpatch with the variable name *movement*, which is described in section 4.3.4.

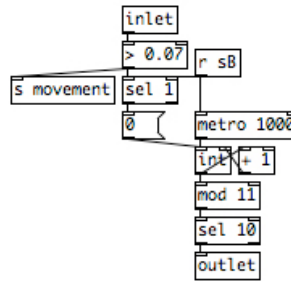


Figure 18. Deciding if the user stands still or not by comparing the vertical force with 0.07.

The statistics from each current running exercise gets calculated in the subpatch *pd stats*, which is shown in Figure 19. The calculation is done by averaging all the values from when the training starts to when the user stands still again. The time of the exercise is also sent to the GUI from this patch. The *s resetStats* variable is used in the statistics patches for the other parameters as well, the variable is responsible for sending the data to the GUI. The *pd stats* patch is used for the other parameters as well, without the time calculation.

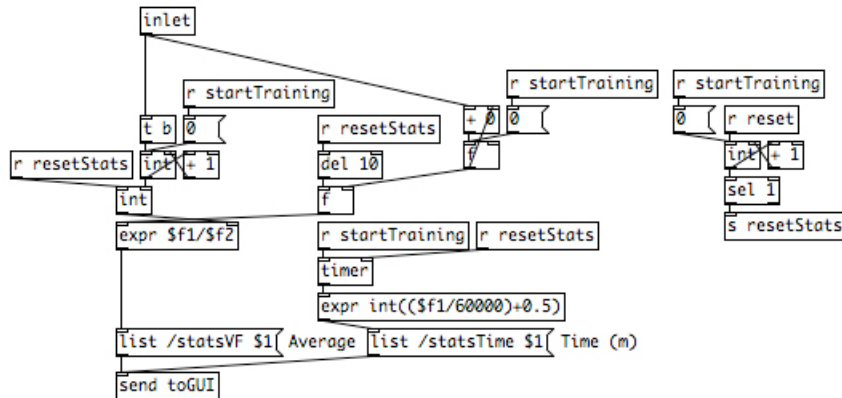


Figure 19. Calculating time and statistics for vertical force.

4.3.4 Step Frequency

The X-axis from the acceleration data is used for detecting steps when running. The subpatch *pd stepDetection* is shown in Figure 20.

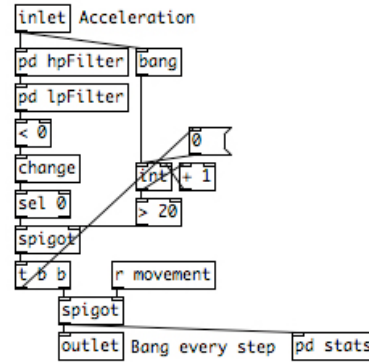


Figure 20. Detecting steps when running.

The step detection starts with a high-pass filter as in the previous equation 4 but with the filtering factor α set to 0.1. Next, a low-pass filter as in equation 3 is used, also with α set to 0.1. The filtering creates a smooth sine wave-like oscillation that can easily be used for detecting steps. When looking at the running gait cycle, the acceleration is zero two times in each cycle. That fact is used in this step detection algorithm, where a “bang” is output whenever the acceleration goes from positive to negative depending on two conditions:

- At least 20 samples, i.e. 200 ms, has to pass between every detected step
- The vertical force has to be bigger than 0.07, otherwise the user is regarded as standing still

The second condition comes from the *r movement* object, which is sent from the vertical force calculations as described in section 4.3.3. The *pd stats* subpatch calculates the number of steps in each running exercise and sends the data to the GUI. After detecting the steps, a calculation of the step frequency is made in the subpatch *pd stepFrequency*, shown in Figure 21.

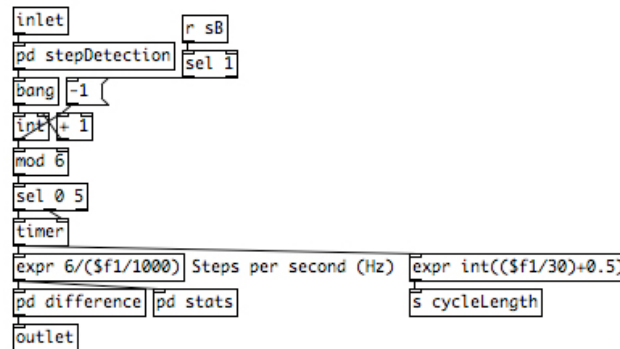


Figure 21. Calculating the step frequency and the length of two running gait cycles in samples.

The step frequency gets calculated by averaging the time of each step during a six-step interval with the output given in steps per second (Hz). A calculation of the time it takes for two steps is also made and sent through the *s cycleLength* object. This number is used for the window size in the RMS calculation of the vertical force, as previously as previously described, and for averaging the other parameters as will be described in section 4.3.5 and 4.3.6.

The step frequency then gets compared to the first 30 seconds of running, just as for the vertical force parameter. Either the step frequency parameter or the velocity parameter can be chosen when training because they are mapped to the same auditory dimension.

4.3.5 Velocity

The X- and Z-axes from the accelerometer go into the subpatch *pd velocity* to calculate a property that was thought to be proportional to the velocity of the runner. Due to the accelerometer drift it is not possible to simply integrate the acceleration to get the velocity. Dubus & Bresin (n.d.) presented a method for computing an approximation of velocity fluctuations around the average velocity. A similar method was used in this project when computing the velocity property for running, with some extensions. An overview of the velocity calculation, which is made in the *pd velocity* patch, is shown in Figure 22.

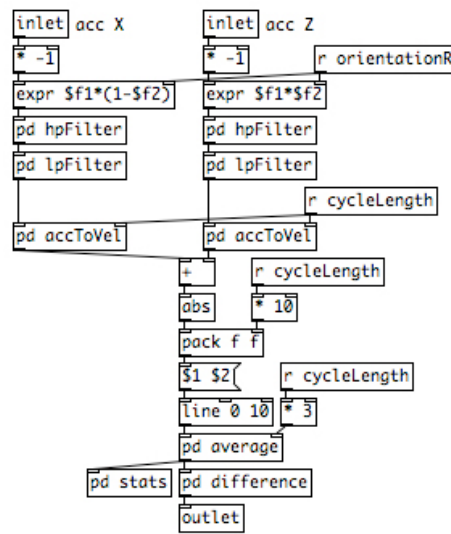


Figure 22. Overview of the calculation of velocity fluctuations using data from the X- and Z-axis of the accelerometer together with the roll angle from the device motion data.

The acceleration data first gets converted to only measure the acceleration in the horizontal direction, with positive values for forward acceleration and negative values for backward acceleration. This is done by taking the tilt of the body into account, which is received from the *r orientationR* object. These values are then filtered in the same way as for the step detection to get a smoother data stream and disregard the gravitational force. The filtered values go in to the *pd accToVel* subpatch together with values received from the *r cycleLength* object, which is the number of samples for two steps. The subpatch *pd accToVel* is based on the patch that Dubus & Bresin used for calculating velocity fluctuations and is shown in Figure 23.

The resulting data from the *pd accToVel* subpatch is shown in Figure 24, where a change in velocity fluctuations can be seen just before the 1500 samples mark when the velocity of the runner was increased.

The absolute value from the velocity fluctuations were then again filtered to get a moving average value from the last running gait cycles. An average value of each cycle was computed and compared to the average value from the first 30 seconds of running. Although this velocity value is not the true velocity, it was decided to use it and test if velocity fluctuations can be used as an indication of velocity.

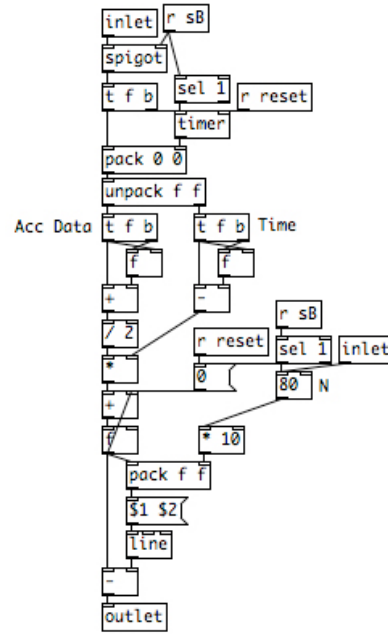


Figure 23. Integrating the acceleration data and removing its moving average value.

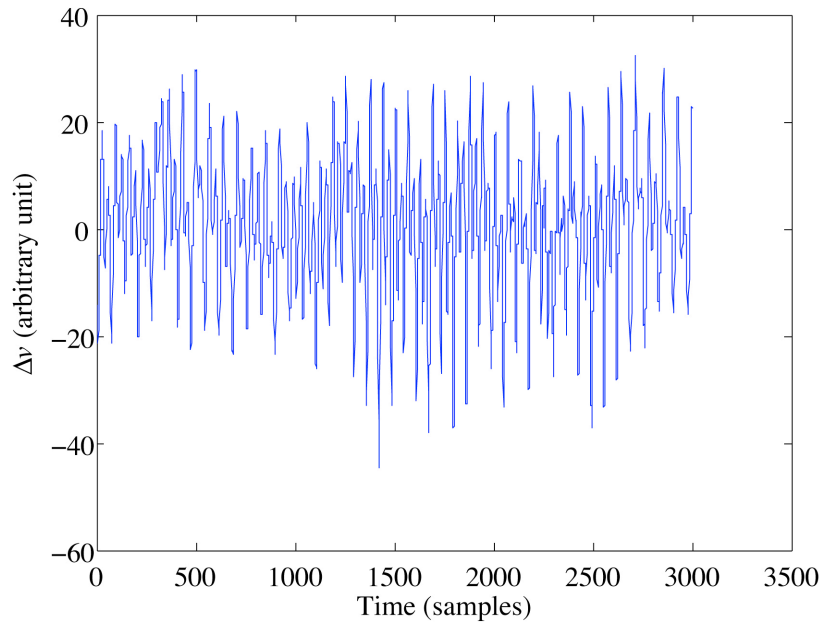


Figure 24. Velocity fluctuations during running. An increase of velocity was made at the 1500 samples mark, indicating that the velocity is somehow proportional to velocity fluctuations.

4.3.6 Body Tilt

The body tilt gets calculated in the subpatch called *pd bodyTilt*. The input to the calculation is the roll angle from the device motion data. The patch is shown in Figure 25. The body tilt gets averaged over a four-step window because the tilt is not constant during the running gait cycle. The length of the window comes from the *r cycleLength* object, which is sent from the *pd stepFrequency* patch. The initial window size is set to 80 samples, i.e. 800 ms. The averaged value then gets compared to the average body tilt from the first 30 seconds of running.

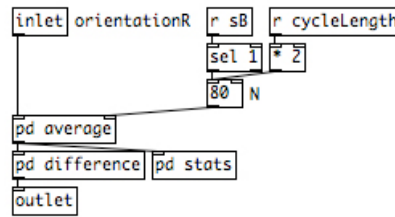


Figure 25. Calculating the body tilt by averaging the roll angle from the device motion data over two running gait cycles.

Since the conversion of the angle of the phone has been done already in the *pd orientation* subpatch as described in section 4.3.2, a value of zero means that the phone is parallel with the surface of the Earth and a value of one means that the phone is perpendicular to the surface of the Earth.

4.4 Sonifying the Body Movements

4.4.1 The Music Player

A music player that loops a selection of ten songs was made by reading the stereo files one after one into two arrays, i.e. one for the left channel and one for the right channel, as shown in Figure 26.

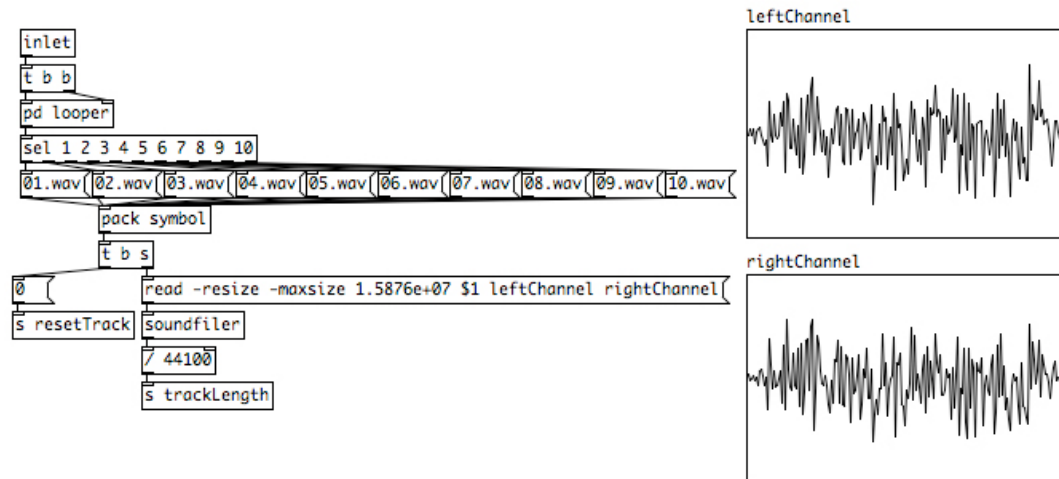


Figure 26. Looping ten sound files and reading them into two arrays.

The songs for the prototype were ten songs from various top lists to insure that the users had heard them before and knew how they should sound. The format of the sound files has to be WAVE, AIFF or similar due to the nature of the *soundfiler* object. The subpatch *pd looper* counts from one to ten and selects which sound file to play. It also receives messages from the on/off button in the GUI so that the song pauses if the button is “off”.

4.4.2 Mappings

To let the users know if their running gait is correct, the music that they are listening to is altered in ways that are intended to be easy to understand. By altering music that the users know how it should be played was expected to give a short response time for the users to change their running gait accordingly. A natural mapping between the physical properties and the auditory dimensions was aimed for in the sonification. As described by Dubus &

The conversion of the velocity values to tempo values is shown in Figure 29. A threshold is set so that when the difference is between -0.075 and 0.075, no tempo shifting is done. A square root function is used so that the tempo shifts more the closer to the threshold the user runs.

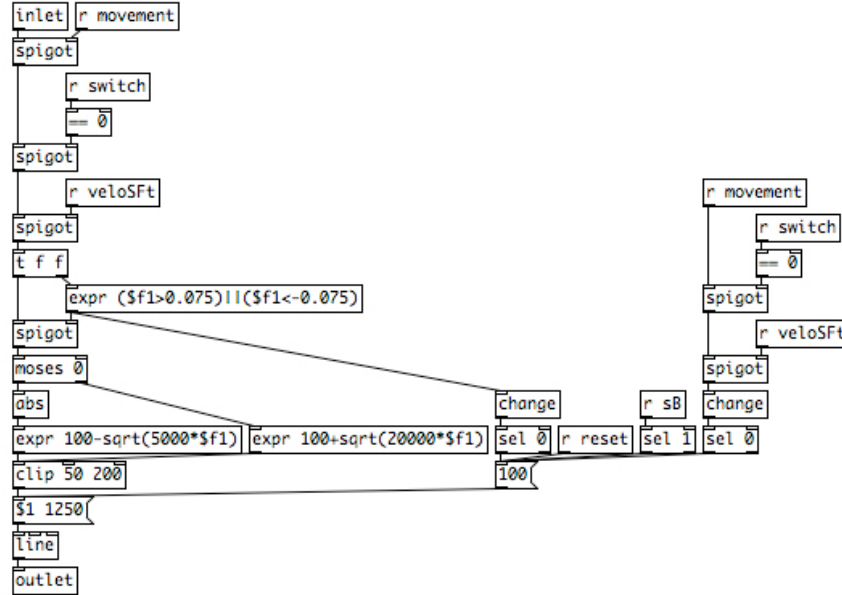


Figure 29. Converting velocity values to tempo values.

A similar conversion of vertical force values to pitch values is shown in Figure 30. The threshold is set to be between -0.02 and 0.02 and a square root function is used for this mapping as well.

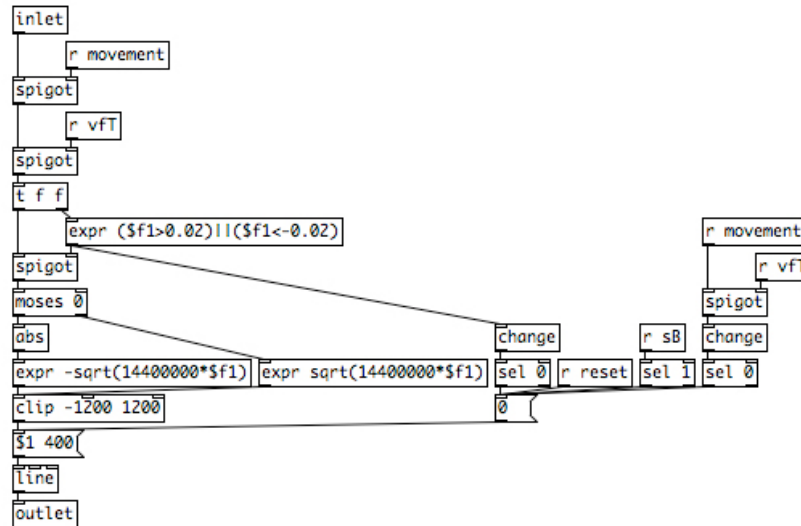


Figure 30. Converting vertical force values to pitch values.

The conversion of body tilt values to cutoff frequency values of the filters is shown in Figure 31. The threshold is here set to be between -0.02 and 0.02 and a logarithmic function is used because of the way humans perceive frequencies of sound waves. The high-pass filter can reduce everything below 2500 Hz and the low-pass filter can reduce everything above 160 Hz so that the music is audible no matter how much or little the user is tilting the body.

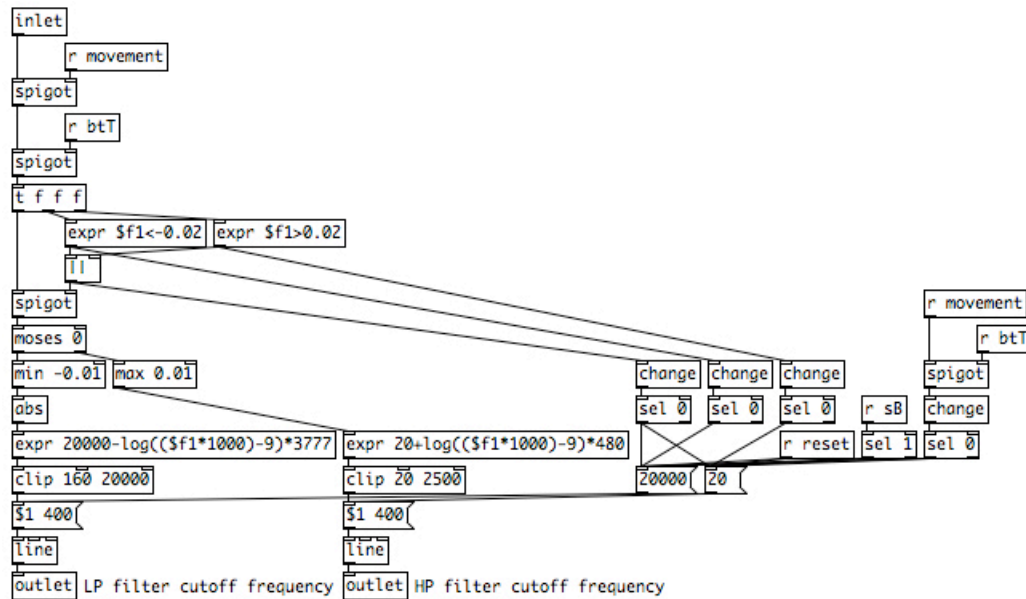


Figure 31. Converting body tilt values to filter cutoff frequency values.

In addition to the interactive sonification, two auditory icons are used. The sound of a starter pistol is used to symbolize that the user should start to run and the sound of a whistle is used to symbolize a coach that is blowing the whistle to tell that the training, i.e. the sonification, starts. The implementation of the auditory icons is presented in section 4.4.5.

4.4.3 Time Stretching and Pitch Shifting

The starting point for the time stretcher and pitch shifter was an example in Pd-extended from *Help/Browser.../Pure Data/3.audio.examples/B14.sampler.rockafella.pd*. The patch makes use of a granular method that allows for pitch shifting without changing tempo and vice versa. Some modifications were done to make the patch fit the context of this application. The subpatch is called *pd timeStretchPitchShift* and is shown in Figure 32.

The input to the patch is given as tempo in percentage of the original tempo, transposition in cents, e.g. one octave is 1200 cents, and the window size of the grains. The principle of this granular method is to read the sound file from an array from start to finish with set tempo. Small grains, or chunks, are then read at different speeds depending on the set tempo shift, pitch shift and window size (here set at 25 ms) of the grains. The playback is divided in two, and both of them are enveloped with a cosine function where one is phase-shifted half a period, for a smoother sound.

The *pd timeStretchPitchShift* patch works together with the *pd musicPlayer* patch to play the music and they communicate with each other through a number of send and receive objects that decide e.g. how long the song is, when to play it and when to change to the next song.

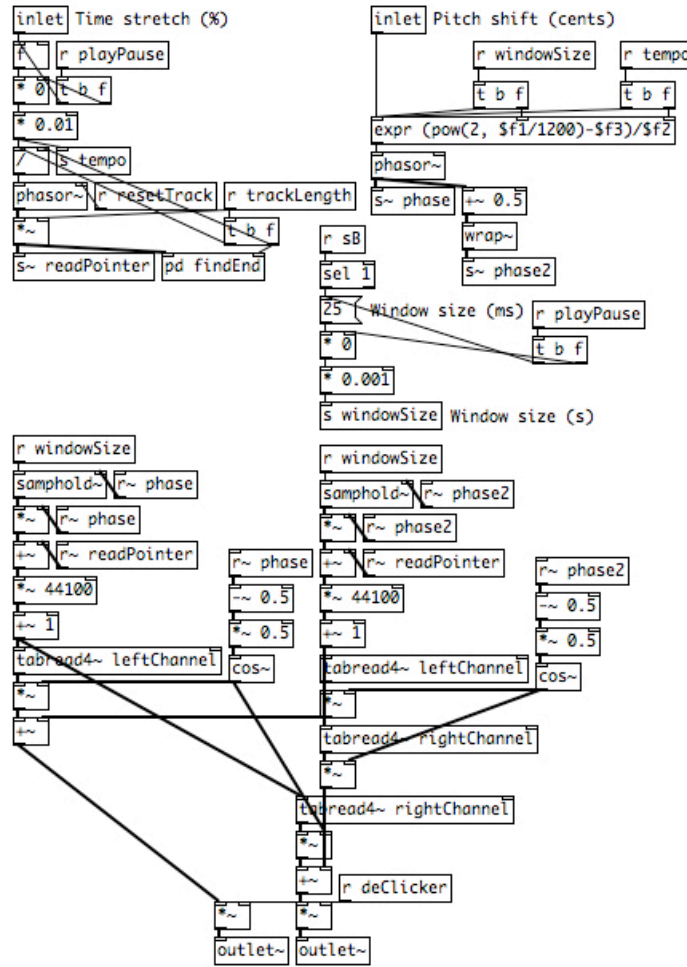


Figure 32. Time stretching and pitch shifting through a granular method, the inputs are set with percentage of original tempo (50 – 200 %) and pitch shifting in cents (-1200 to 1200 cents).

4.4.4 Filters

In the *pd filters* subpatch, the initial setting is a low-pass filter set at 20 kHz and a high-pass filter set at 20 Hz which is considered to be the average human range of hearing. The *pd filters* subpatch is shown in Figure 33.

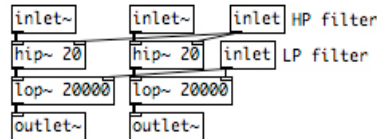


Figure 33. High-pass and low-pass filters for the left and right channels of the sound files.

The converted body tilt values are sent to the inlets to set the cutoff frequency of the different filters. Since the filter objects in Pd works in mono two of each filter had to be used, one for the left channel and one for the right channel.

4.4.5 Auditory Icons

Two auditory icons were used, which is shown in Figure 34. The left inlet receives a “bang” through the *r reset* object and the right inlet receives a bang through the *r startTraining* object.

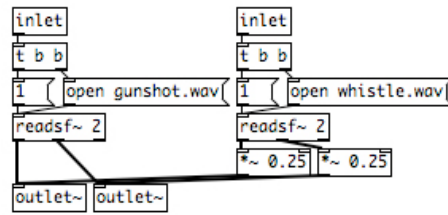


Figure 34. Auditory icons. The gunshot.wav file is played when the application gets reset and the whistle.wav file is played when the training, i.e. the sonification, starts.

When the vertical force is low enough the user is regarded as standing still and the sound of a gunshot is played. The sound of a whistle is played after the average computations have been made during the first 30 seconds on running.

4.5 Sound Examples

A video was made to demonstrate how it could sound like when running with the application. All the parameters including the auditory icons are included, both individually and combined, together with explanations on why the sound is changing the way it does. The video is found on the following URL:

- <https://vimeo.com/99441665>

5 Results

This chapter presents the results of the user tests. A validation of the step detection algorithm was done and the application was tested in different ways to see how it behaves together with users in different situations.

5.1 Validation of the Step Detection

Tests were conducted to validate the step detection in the application. A step counter was implemented, which counted all the detected steps, as well as a reset button that could be pressed to set the counting to zero. The tests were made with five participants that had the mobile phone attached to the body in the *Adidas Media Belt X14341* as previously mentioned. In addition, a pedometer from Silva called *Silva ex step* was attached to the belt. The participants ran for some minutes each and the results are concluded in Table 1.

Participant	Step Count (App)	Step Count (Silva ex step)	Difference (% Δ)
1	713	709	0.564
2	363	364	- 0.275
3	545	539	1.113
4	381	377	1.061
5	586	580	1.034
All	2588	2569	0.740

Table 1. Results from the validation tests of the step detection algorithm.

The results showed that the application detected fewer steps than the *Silva ex step* pedometer for four out of five tests. The total difference between the two was 0.740 %.

5.2 Experiments

The application was tested in different outdoor environments with runners of varying levels of experience, both on hilly terrains and on flat asphalt, to see how the application worked in different situations. Three participants conducted the tests. The first participant (P1) was a male with a background in sports. The second participant (P2) was a female with less experience of sport activities and being only an occasional runner. The last participant (P3) was a male with a longer background in sports including athletics.

Before the tests, the function of the application together with the different mappings was explained. As written in section 4.4.1, ten songs from different top lists were used so the participants knew how the music sounded when played correctly. An addition to the application was implemented, where a choice could be made to make the music silent for the first three minutes of running to see how the running technique changed when the sound came back. The test procedures for the three participants are explained below:

- P1 ran in a mixed terrain, both hilly and flat, first with the music muted for three minutes, then with sonification for the remaining part. The settings were made to

either increase or decrease the different parameters compared to the first 30 seconds of running.

- P2 ran back and forth on a long and flat surface of asphalt. The ground had a slight slope, so half of the running was made downhill and half uphill. Each running exercise was two to three minutes long and the settings were made to keep the values from the first 30 seconds of running.
- P3 ran with the same settings as P2, partly on the same asphalt section, but also on a flat section of gravel so the total distance was approximately doubled.

The four parameters were first tested individually and then with three of the parameters combined. After each running exercise, unstructured interviews with the participants were made to see how they perceived the sound and how well they thought they were able to change their body according to it. The data from the tests were saved in text files and are presented in the coming sections together with the participants' thoughts of the application. The saved data starts from when the training starts, i.e. the data from the first 30 seconds of running were not saved.

5.2.1 Vertical Force

The setting for the vertical force test with P1 was made to decrease it with 6 %. The results of the running exercise is shown in Figure 35. The sonification starts around the 150-second mark.

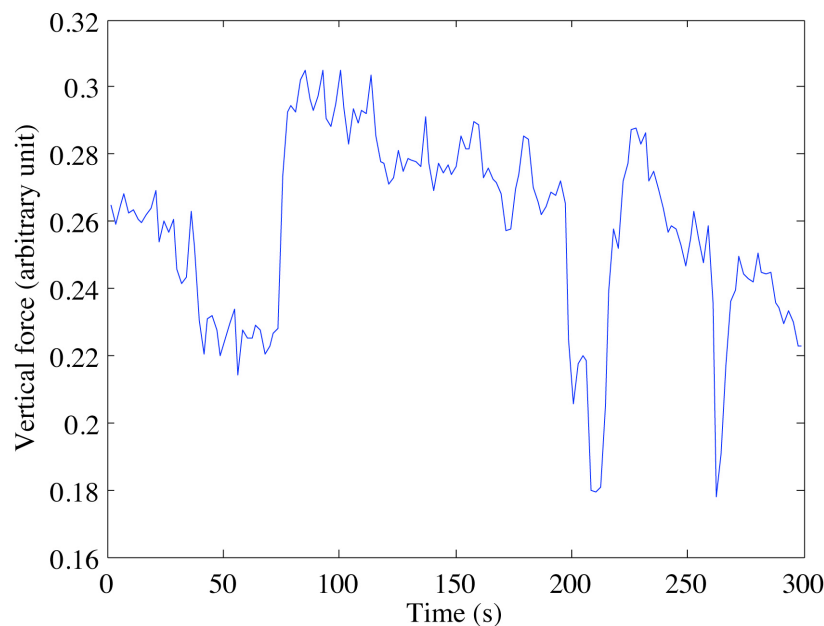


Figure 35. The vertical force of P1. He was over the threshold for most of the time, but managed to decrease the vertical force during the latter part of the exercise when the sonification started.

During the interview, P1 said that the pitch was too high when the music came back and he was trying to reduce the upward movement in the running. He said that the music could be heard in its original form towards the end of the running, when he tilted the body more and focused the force away from the vertical direction.

The vertical force test for P2 is shown in Figure 36. P2 said that the pitch of the music was good during the downhill running, which was when the average value was computed. She said that the pitch was too low on the way back, but that it was hard to increase the

force due to the uphill, although she understood why the music had changed the way it did and what she would have had to do to get it right.

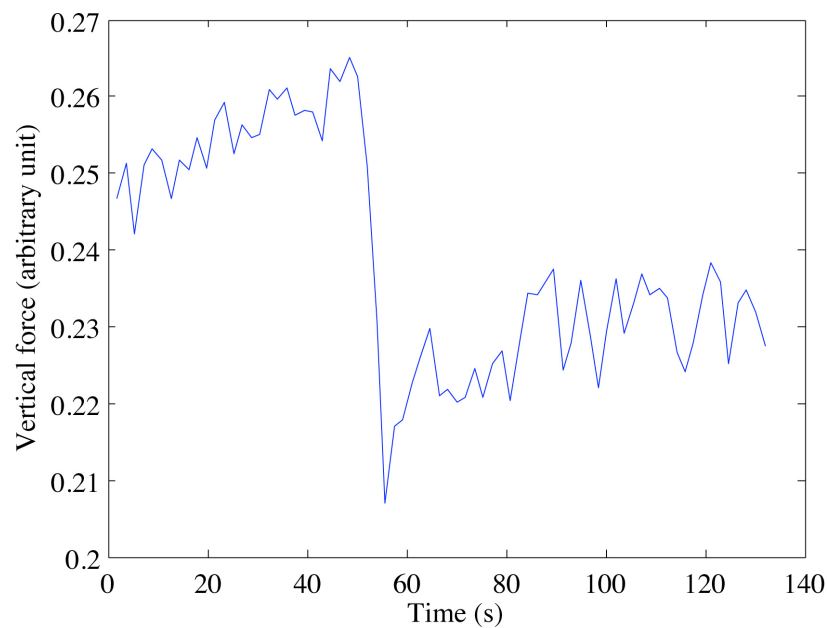


Figure 36. The vertical force of P2. She managed to keep the vertical force in the beginning but due to the uphill during the second half of the running, her vertical force was too low and she was not able to increase it.

The vertical force of P3 can be seen in Figure 37.

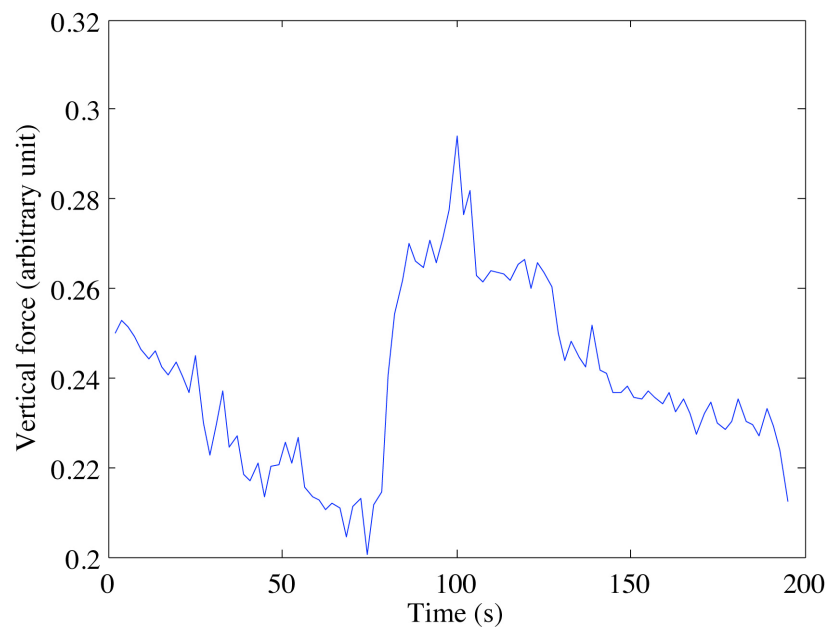


Figure 37. The vertical force of P3. He managed to correct the vertical force around the 100-second mark, otherwise it was too low.

P3 said that it was hard to adjust the vertical force and the pitch was too low for most of the exercise. He said that the music was played normally after half the distance, which can be seen in the figure. Then the music was too low during the uphill so he tried to increase the vertical force but thought it was hard.

5.2.2 Step Frequency

The settings for P1 during the step frequency test were made to increase the frequency with 4 %. The results are shown in Figure 38. P1 said that it was easy to change the step frequency when listening to the music. When the music came back after being muted he heard that the tempo was too slow so he increased the step frequency and could hold on to it for the remaining part of the exercise.

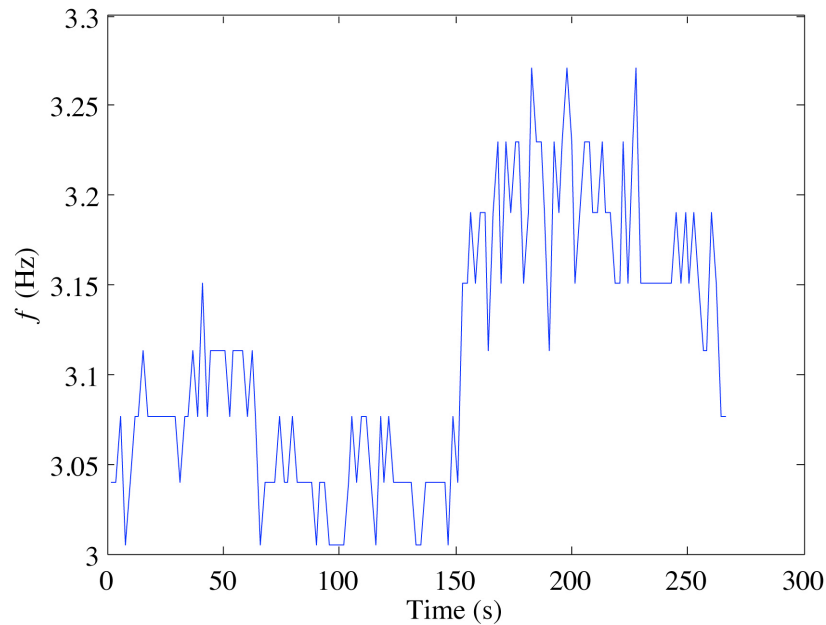


Figure 38. The step frequency of P1. The music was muted until around the 150-second mark, where a distinct increment of his step frequency can be seen.

The results of P2 can be seen in Figure 39.

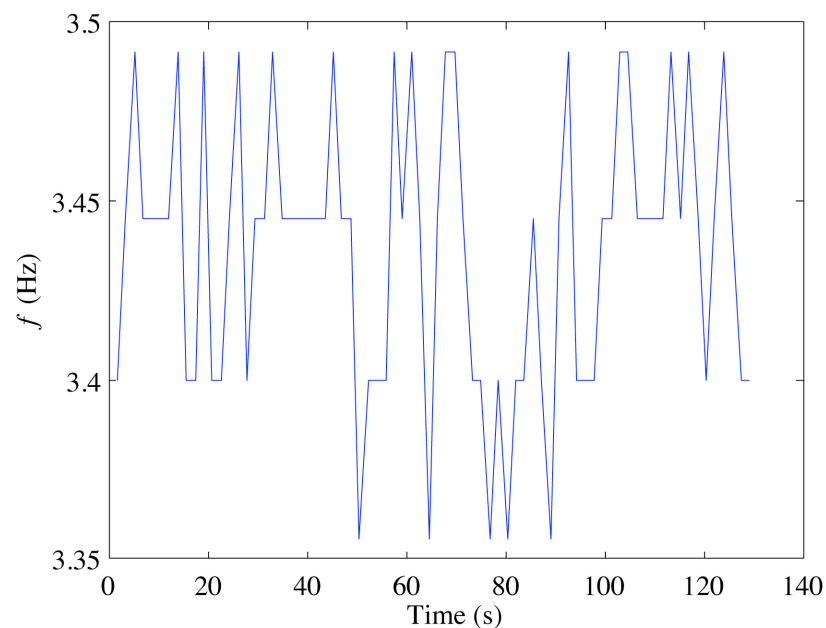


Figure 39. The step frequency of P2. A somewhat constant step frequency was held during the whole exercise but it went down in the middle sometimes.

P2 said the tempo of the music was normal for the most of the running exercise, but during and after the turn on her way back uphill the tempo was occasionally slower. Then she increased her step frequency and the music went back to normal again. She said that it was clear how the music behaved in relation to her motion.

The step frequency of P3 is shown in Figure 40. P3 said that it was easy to adjust the step frequency. He said that the music went faster and slower sometimes but he managed to adjust it every time. Furthermore, he said that when the uphill started, the music became slower so he increased the step frequency.

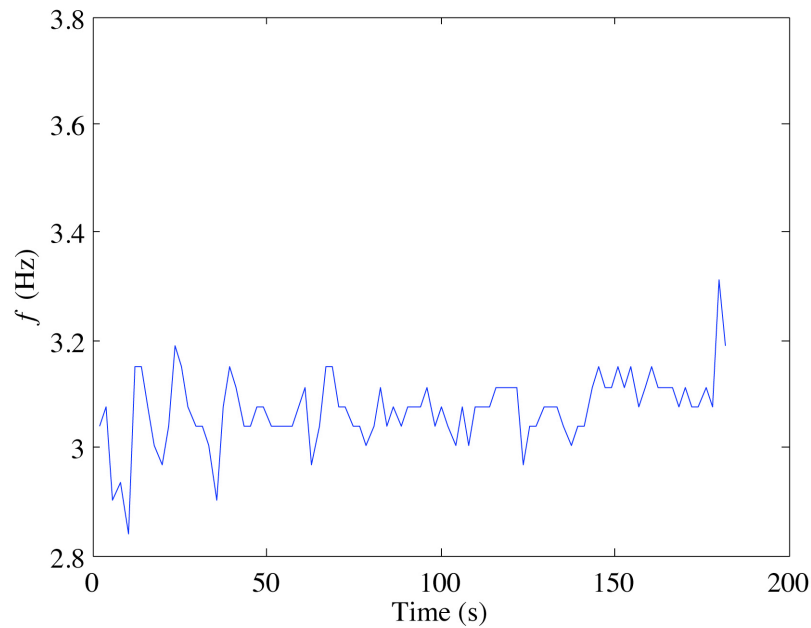


Figure 40. The step frequency of P3. Some fluctuations can be seen in the beginning but then he managed to keep a more constant step frequency.

5.2.3 Velocity

The absolute value of the velocity fluctuations was tested to see if it can be used as an indication of the velocity. The settings for P1 were made to increase the velocity with 8 %. The results are shown in Figure 41. P1 said that the tempo of the music was too slow when the sonification started so he increased the speed a little bit. At first, the result of that increment did not affect the music much, but after a while the tempo became too fast. Then he tried to slow down a bit but the music kept being too fast except for some short moments when it was played back in normal speed. He said that the music changed rapidly sometimes although his own speed did not change that much.

The results from P2 are shown in Figure 42. She was jogging in a slow pace and a change can clearly be seen during the uphill before the 100-second mark. She said that the tempo of the music was too fast almost all the time and did not feel that it was corresponding to her own velocity changes.

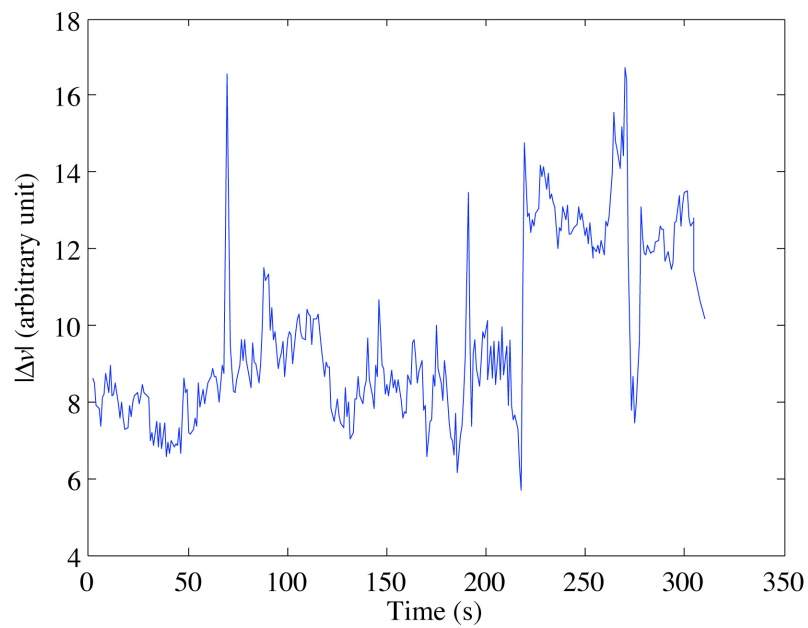


Figure 41. The absolute value of the velocity fluctuations used as an indicator of the velocity of P1. The graph indicates that the velocity did increase but the user said that the music was too fast for the most part after the sonification had started even though he slowed down.

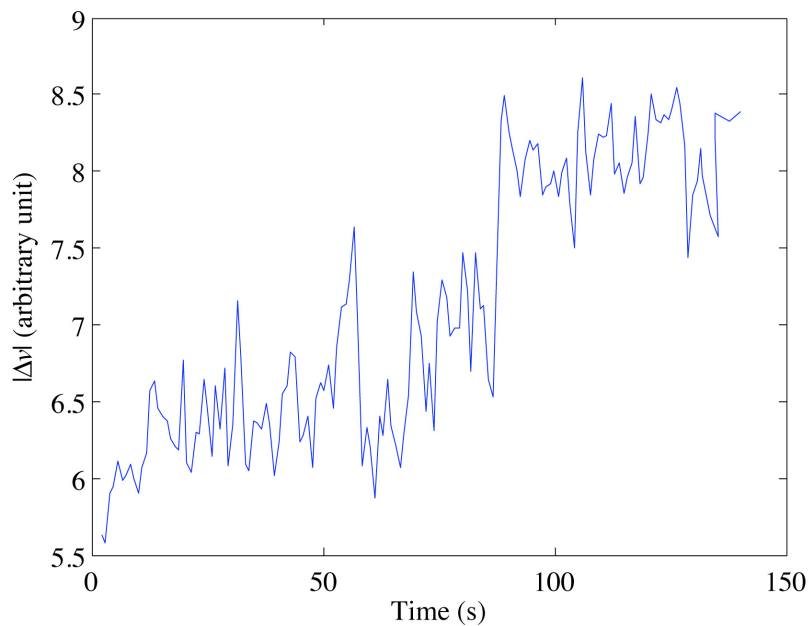


Figure 42. The absolute value of the velocity fluctuations used as an indicator of the velocity of P2. A distinct increase in velocity fluctuations can be seen in the uphill running during the second half of the exercise.

The results of P3 are shown in Figure 43. P3 said that the tempo of the music was too fast immediately after the whistle sound. On the flat gravel ground the tempo was normal but during the uphill it went down.

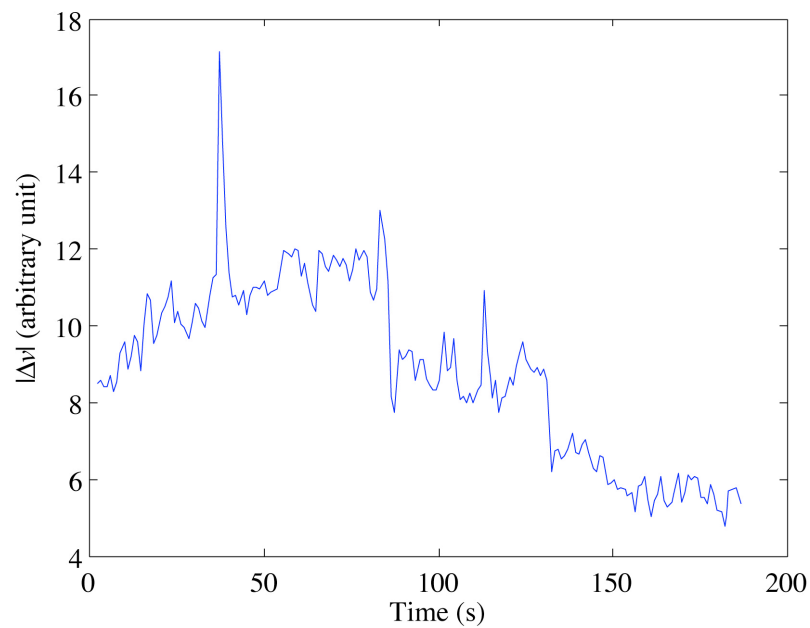


Figure 43. The absolute value of the velocity fluctuations used as an indicator of the velocity of P3. During the downhill the velocity increased, on the flat ground it was somewhat constant and during the uphill it decreased.

5.2.4 Body Tilt

The results from the body tilt test of P1 are shown in Figure 44. The settings were made to increase the body tilt with 4 %.

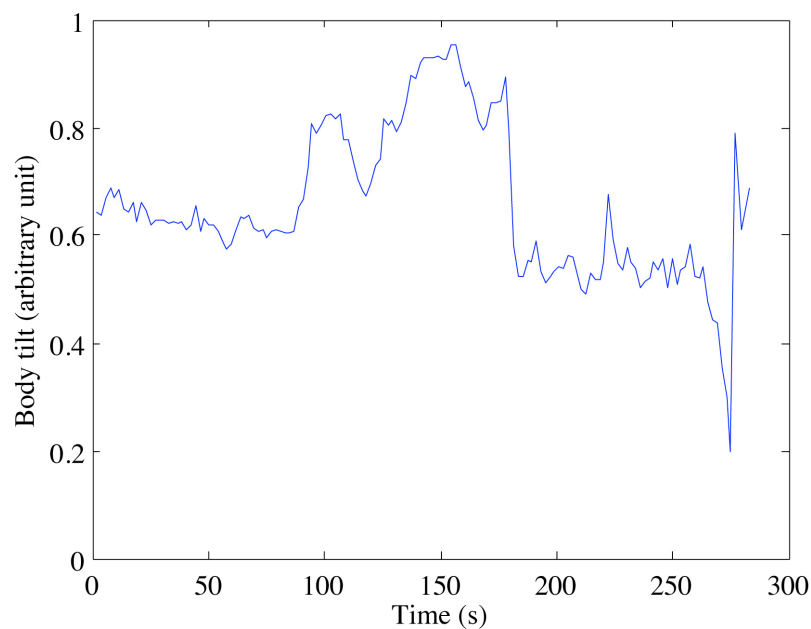


Figure 44. The body tilt of P1. After running rather upright in the middle, he managed to increase the body tilt during the last third of the exercise.

P1 said that he got tired in the middle of the exercise and during a downhill the music came back without any bass in it. He then tried to increase his speed but was not able to

restore all of the bass in the music, but by tilting only the upper body he managed to get the music to sound right.

The results from the body tilt test of P2 are shown in Figure 45. P2 said that the bass was missing in the music for most of the time. She did not manage to run faster and it is clear that her body tilt decreased more and more during the uphill towards the end.

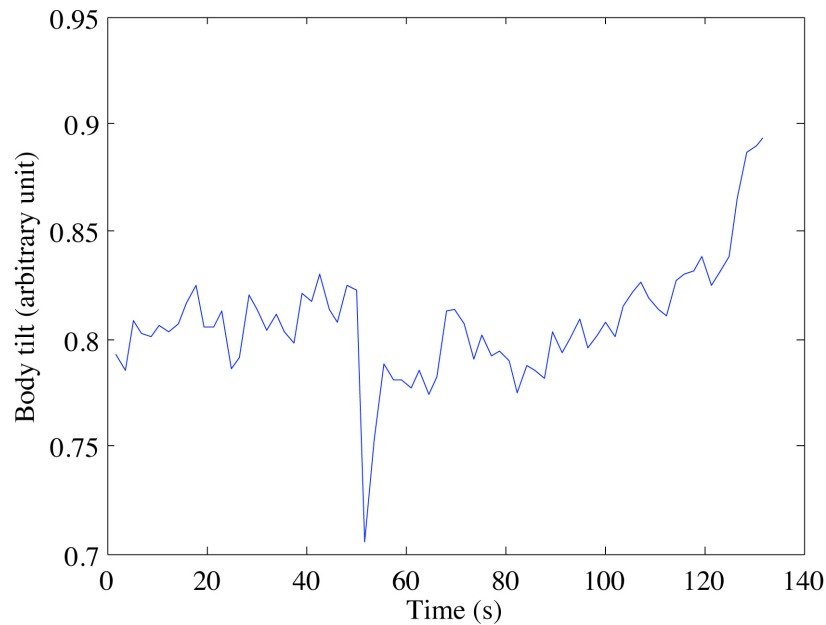


Figure 45. The body tilt of P2. The body tilt decreased more and more during the second half of the exercise.

The body tilt test of P3 are shown in Figure 46.

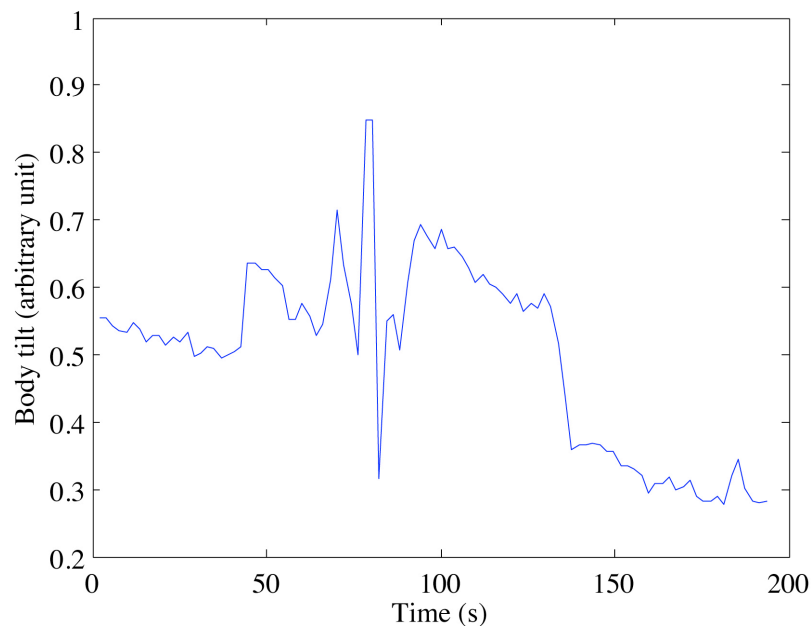


Figure 46. The body tilt of P3. The body of his was tilted too much for most of the exercise, especially towards the end of the uphill. However, he did manage to get it right for some time.

P3 said that the treble was missing for the most parts of the running although the music was played correctly during the middle of the exercise. On the way up, the treble was missing all the time. He said that he tried to adjust the tilt but did not think about the delay of the feedback.

5.2.5 Perception of Combined Parameters

All parameters except velocity were tested together to see how well the participants could extract information from the music. The data from this test was not saved because the aim of the test was not to optimize the running technique but simply test if they could control the sound with their movements and if the mappings felt logical. The settings were made to keep the values from the first 30 seconds of running. All participants thought that the auditory icons worked well, they understood when to start to run and when the sonification started. Presented below are the noteworthy thoughts from the participants.

P1 said that the vertical force and body tilt changed together. He said that, for some ten seconds the pitch was too high and the bass in the music was not apparent so he increased the speed, thus tilting the body forward, and the music went back to normal again. However, he said that it was hard to focus on more than one thing at a time but the mappings were good and he understood how the body movement should change for the music to change.

P2 said that it was hard to understand all the dimensions of the sound and that she probably would need some more training to be able to control the music completely. Then she said that the bass was removed immediately after the whistle was blown so she tilted her body forward a little bit more and it came back. After a bit of uphill the tempo of the music decreased so she increased the number of steps so the music went back to normal. Lastly, she said that the pitch of the music was correct all the time so she did not have to change the body according to that parameter.

P3 said that he managed to get the music right during flat surfaces. When running on an uphill, his step frequency was too high. He showed a clear understanding of how the sonification worked but said that it was hard to adjust the body according to it during distances that were not flat.

6 Discussion

This chapter provides a discussion based on the results of the project and its relation to previous work that have been done in the research areas of biomechanics and sonification.

A mobile application for long-distance running that addresses the users' running gait has been implemented in Pd together with MobMuPlat. By helping the users to understand their body movements through sonic interaction, the aim was to make them be able to change their movements in relation to the music they are listening to while running. A critical part of the running gait analysis was related to the running gait cycle, as a running exercise is repetitive in its nature. Therefore, a step detection algorithm was needed for analyzing the other parameters that were chosen for the application. A validation of the step detection algorithm was made by comparing the results from the application with the pedometer *Silva ex step* to be sure that the step detection worked. The difference between the two was 0.740 % and the most likely reason for that was that the application is unstable when the user stands still. When standing still, steps are detected because of the small positive and negative accelerations. By turning the step detection off when the vertical force is beneath a certain threshold, the number of falsely detected steps gets minimized. However, to calculate the vertical force needs some seconds and during that time, steps are added in the step counter. This is not a problem during running because the crossing of zero acceleration only occurs two times during each running gait cycle, which provides a reliable number of step detections.

Four parameters relating to running gait were implemented and three of them proved to give successful results regarding the sonification. During the implementation phase, velocity fluctuations around the average velocity were thought to work as an indicator of the true velocity, however this solution did not work out well during the tests. It may be true that the velocity fluctuations are bigger during sprinting than jogging, but jogging in different velocities did not seem to correspond to the varying velocity fluctuations for the test participants. A question that was raised during the implementation phase of the velocity parameter was regarding the delayed feedback of the sonification. If the signal is filtered a lot by averaging it over a longer period of time, the sonification gets more reliable but it also gets more delayed. That can be worked through by having a larger tolerance for how much the user can be above or beneath the desired threshold, which was made for the velocity parameter. It can be seen in the figures from the test regarding velocity that it was fluctuating a lot.

Since velocity is a key component of running performance, this area would benefit from a robust solution. Perhaps a GPS solution is the best alternative today, but it does not work in an indoor environment. Another way could be to use the oscillating velocity fluctuations as Dubus & Bresin did in an application for elite rowers, but this would need another sound model than in this project. Since it would make the tempo constantly increase and decrease with the current sound model, it would probably not be pleasant to listen to. If a sufficient velocity solution is found, mapping it to the tempo of the music seems like a good idea, according to the test participants they knew how they in theory would have to change their velocity to change the tempo of the music.

The other three parameters did however work out well. The calculation of the vertical force seemed to be robust for all participants. All participants talked as if the body was connected to the sound, which points to the fact that a natural mapping was made.

Natural interactions between the users and the sound are regarded as important and it was indeed an aim of the project. The participants showed a good understanding of why the sound changed the way it did. By understanding the mapping and being able to change the body accordingly, the vertical displacement of the user's center of mass will decrease and that is a key for improving one's RE. Another central aspect in running technique is the tilt angle of the body, the body should be tilting forward from the ankles and upward. The more it tilts the faster one can run, as a consequence of the lowering of the center of mass. This feature was implemented as the body tilt parameter, and was also successfully mapped to filters. The test participants understood well how the mapping worked and how they should have changed their tilt angle by listening to the music. For all the participants, the results were however not convincing that they managed to alter their movements. E.g., P1 said that he tilted his upper body to get the music to sound right. By tilting only the upper body, it has the reverse effect and makes the running less efficient. This may have been the result of having the increment set too high. The tilt of P3 almost indicated that he ran perpendicular to the ground during the uphill; this was most likely a consequence of the phone's placement or running with a slight swayback. That is not a problem however, since only the difference in the tilt is used and not its real value. The body tilt results showed that the tilt changed much depending on whether the surface was flat or hilly, which can be a problem when using this application in an outdoor environment that is not flat.

The step frequency parameter was probably the most successful one in the user tests. All participants managed to adjust their step frequency to get the music being played in its correct tempo. By being able to control and analyze one's own step frequency, different effects may occur. E.g. when runners decrease their step frequency, it is said that they get a better RE due to the fact that they have to overcome gravity a fewer number of times. But it is also said that elite runners most often have a higher step frequency than casual runners. Either way, it is good to be able to have a consistent step frequency. An interesting addition to the application could be to analyze the tempo of the music beforehand and adjust it to the user's desired step frequency; thereafter the users can simply synchronize their steps with the music's beats. A problem in this solution is that the users may want to listen to their favorite music as it is meant to be heard, and played back in its correct tempo.

One surprising finding during the tests was the participants' ability to analyze how they had been running. They remembered where the music had changed and had thoughts on why it had changed the way it did, which points to a strong connection between the body movements and the sound. The concept of sonification should provide the user a deeper understanding of a process by simply listening to it and the interactivity in this application did really provide deeper analysis abilities for the test participants. As Tucker wrote, to learn a good running technique is done in three stages. With this application, the second and third step could be merged, by practicing and getting instructions at the same time. As it is common to listen to music during running, to sonify it can be a part of the users' exercise and through it they can train, get instructions in real-time and instantly know if they have improved.

7 Conclusions and Future Work

This chapter provides a summation of the most important findings of the project together with some suggestions for future work.

7.1 Conclusions

It has previously been shown that sonification is of great use together with sports applications and this project has strengthened that belief. The aim of the project was to implement a mobile application that makes use of interactive sonification to improve the user's running technique through a better body posture and movement. By improving the running technique, the runner consumes less energy and can thus make a better performance.

Four parameters were tested and three of them showed good results. More work has to be done for calculating the velocity using sensors to provide a real-time feedback of it. Another alternative is to sonify the changes of the acceleration, as Schaffert et al. did with results that indicated an increment of the velocity.

The runner's vertical force, body tilt and step frequency were however implemented and works as it was suggested beforehand. The application proved to have a functional mapping between physical and auditory dimensions that make the user understand how to alter the body movement. The application was successfully tested on three runners with different levels of experience as sportsmen. All of them understood how the body movements related to the sonic feedback but more practice of running with the application could be helpful to be able to change the body posture and be in control of the running technique.

7.2 Future Work

Due to the fact that all of the information regarding the running gait was extracted from one mobile phone that was placed on one place of the user's body, some important aspects that would have extended the application's gait analysis were left out. Possible extensions could be realized by having more sensors placed on other parts of the body. There are, e.g., the sensors from Texas Instruments and x-io Technologies that can communicate with a mobile phone via Bluetooth or Wi-Fi. Another future possibility could be to track a complete skeleton of the user through a number of sensors placed on the body. Additional parameters relating to the feet and arms could then be implemented, such as the landing of the foot, which should be on the rearfoot as close as possible to the ground projection of the center of mass. Another parameter could be related to the arms, when bending them e.g. 90 degrees, it is possible to run faster. The question is how many dimensions that can be sonified at the same time and still having it to make sense for the user.

All participants during the experiment in this project said that they had problems focusing on more than one auditory dimension at a time. Although they managed to alter their body movements to some extent by, e.g., realizing that the parameters are connected in some ways. A skeleton tracking with sonification would perhaps benefit from another sound model, but there are more possible implementations that can be done in the chosen one. Spatialization could be used to separate between movements on the left and right side of the body by panning, filtering or delaying one of the two channels.

To sonify body movements has great potential for further use. As Crowell et al. did with visual feedback, sonification could be used to decrease the vertical force for reducing the risk of getting stress fractures. It can also be used to get a better body posture when walking, by e.g. having two sensors placed on the back to reduce the tilt of the upper body. Another interesting extension would be to add haptic feedback from e.g. the Notch sensors.

It is clear that mobile devices have gotten more powerful during the last decade and it allows for even more complex sound models. To use Pd together with MobMuPlat has proven to be a good platform for prototyping. There is only a question of time before commercial applications of this kind arises as it has been shown that both less and more experienced recreational runners can make use of sonification. A possible commercial application could be implemented with libpd and e.g. make use of the user's own music library or a streaming service to access music.

Bibliography

- Barrass, S., Schaffert, N. & Barrass, T., 2010. Probing Preferences between Six Designs of Interactive Sonifications for Recreational Sports, Health and Fitness. In R. Bresin, T. Hermann, & A. Hunt, eds. *Proceedings of ISON 2010, 3rd Interactive Sonification Workshop*. Stockholm, Sweden: KTH, pp. 23–29.
- Bolíbar, J., 2012. *Kinect Audio-Runner: Audio Feedback for Improving Performance in Long-Distance Running*. KTH.
- Bood, R.J. et al., 2013. The Power of Auditory-Motor Synchronization in Sports: Enhancing Running Performance by Coupling Cadence with the Right Beats. *PLOS ONE*, 8(8).
- CNMAT, 2011. Introduction to OSC. Available at: <http://opensoundcontrol.org/introduction-osc> [Accessed February 28, 2014].
- Conley, D.L. & Krahenbuhl, G.S., 1980. Running Economy and Distance Running Performance of Highly Trained Athletes. *Medicine and Science in Sports and Exercise*, 12(5), pp.357–360.
- Crowell, H.P. et al., 2010. Reducing Impact Loading During Running With the Use of Real-Time Visual Feedback. *Journal of Orthopaedic & Sports Physical Therapy*, 40(4), pp.206–213.
- Dubus, G., 2012. Evaluation of Four Models for the Sonification of Elite Rowing. *Journal on Multimodal User Interfaces*, 5(3-4), pp.143–156.
- Dubus, G. & Bresin, R., 2013. A Systematic Review of Mapping Strategies for the Sonification of Physical Qualities. *PLOS ONE*, 8(12).
- Dubus, G. & Bresin, R., Exploration and Evaluation of a System for Interactive Sonification of Elite Rowing. *Sports Engineering*, Submitted.
- Edelstam, F. & Eelde, C.-J., 2013. *Mappning mellan indata och utdata i löparapplikation*. KTH.
- Eriksson, M. & Bresin, R., 2010. Improving Running Mechanics by Use of Interactive Sonification. In R. Bresin, T. Hermann, & A. Hunt, eds. *Proceedings of ISON 2010, 3rd Interactive Sonification Workshop*. Stockholm, Sweden: KTH, pp. 95–98.
- Gonowon, J., 2007. The Physics of Efficient Running. Available at: http://ffden-2.phys.uaf.edu/212_spring2007.web.dir/jessica_gonowon/gonowon_page1.html [Accessed January 20, 2014].
- Hermann, T. & Hunt, A., 2005. An Introduction to Interactive Sonification. *IEEE Multimedia*, 12(2), pp.20–24.
- Hummel, J. et al., 2010. Interactive Sonification of German Wheel Sports Movement. In R. Bresin, T. Hermann, & A. Hunt, eds. *Proceedings of ISON 2010, 3rd Interactive Sonification Workshop*. Stockholm, Sweden: KTH, pp. 17–22.

- Hunt, A. & Hermann, T., 2004. The Importance of Interaction in Sonification. In S. Barrass & P. Vickers, eds. *Proceedings of ICAD 04-Tenth Meeting of the International Conference on Auditory Display*. Sydney, Australia: International Community for Auditory Display.
- IEM, 2014. Pure Data - PD Community Site. Available at: <http://puredata.info> [Accessed February 28, 2014].
- Iglesia, D., MobMuPlat. *Iglesia Intermedia*. Available at: <http://www.mobmuplat.com> [Accessed February 28, 2014].
- Kirn, P., 2012. libpd. *Create Digital Music*. Available at: <http://libpd.cc/about/> [Accessed March 3, 2014].
- Kramer, G. et al., 1999. *Sonification Report: Status of the Field and Research Agenda. Report prepared for the National Science Foundation by members of the International Community for Auditory Display*, Santa Fe, New Mexico, USA.
- Mccormick, C., Muddu, K. & Rousseau, A., 2013. PdDroidParty. Available at: <http://droidparty.net> [Accessed May 5, 2014].
- Notch Interfaces, 2014. Notch. Available at: <http://www.wearnotch.com> [Accessed April 16, 2014].
- Novacheck, T.F., 1998. The Biomechanics of Running. *Gait and Posture*, 7(1), pp.77–95.
- Phillips, M., 2013. Improving Your Speed: Step Frequency and Step Length. *Runners Connect*. Available at: <http://runnersconnect.net/running-training-articles/improve-running-speed-step-frequency-and-step-length/> [Accessed May 22, 2014].
- Di Prampero, P.E. et al., 1993. Energetics of Best Performances in Middle-distance Running. *Journal of Applied Physiology*, 74(5), pp.2318–2324.
- Reality Jockey Ltd., 2013. RjDj. Available at: <http://rjdj.me> [Accessed May 5, 2014].
- Rocchesso, D., Polotti, P. & Delle Monache, S., 2009. Designing Continuous Sonic Interaction. *International Journal of Design*, 3(3), pp.13–25.
- Saunders, P.U. et al., 2004. Factors Affecting Running Economy in Trained Distance Runners. *Sports Medicine*, 34(7), pp.465–485.
- Schaffert, N. et al., 2009. Exploring Function and Aesthetics in Sonifications for Elite Sports. In C. Stevens et al., eds. *Proceedings of the 2nd International Conference on Music Communication Science*. Sydney, Australia: University of Western Sydney.
- Schaffert, N., Mattes, K. & Effenberg, A.O., 2010. Listen to the Boat Motion: Acoustic Information for Elite Rowers. In R. Bresin, T. Hermann, & A. Hunt, eds. *Proceedings of ISON 2010, 3rd Interactive Sonification Workshop*. Stockholm, Sweden: KTH, pp. 31–37.

- Suied, C. et al., 2010. Why Are Natural Sounds Detected Faster Than Pips? *Journal of the Acoustical Society of America*, 127(3), pp.105–110.
- Susini, P. et al., 2012. Naturalness Influences the Perceived Usability and Pleasantness of an Interface's Sonic Feedback. *Journal on Multimodal User Interfaces*, 5(3-4), pp.175–186.
- Texas Instruments, 2013. CC2541 SensorTag Quick Start Guide. Available at: <http://www.ti.com/lit/ml/swru324b/swru324b.pdf> [Accessed January 23, 2014].
- Tucker, R., 2007. Running Technique Part II. *The Science of Sport*. Available at: <http://www.sportsscientists.com/2007/09/running-technique-part-ii/> [Accessed January 20, 2014].
- Varni, G. et al., 2012. Interactive Sonification of Synchronisation of Motoric Behaviour in Social Active Listening of Music with Mobile Devices. *Journal on Multimodal User Interfaces*, 5(3-4), pp.157–173.
- Williams, K.R. & Cavanagh, P.R., 1983. A Model for the Calculation of Mechanical Power During Distance Running. *Journal of Biomechanics*, 16(2), pp.115–128.
- Worrall, D., 2009. *Sonification and Information: Concepts, Instruments and Techniques*. University of Canberra.
- x-io Technologies, 2013. x-OSC User Manual Version 0.3. Available at: <http://www.x-io.co.uk/downloads/x-OSC-User-Manual-v0.3.pdf> [Accessed January 23, 2014].
- Yang, J. & Hunt, A., 2013. Sonic Trainer: Real-time Sonification of Muscular Activity and Limb Positions in General Physical Exercise. In *Proceedings of ISON 2013, 4th Interactive Sonification Workshop*. Erlangen, Germany: Fraunhofer IIS.

